

The Long-Term Design of Electrical Power
Distribution Networks

by

T. Lang

Thesis presented for the Degree of Doctor of Philosophy
of the University of Edinburgh in the Faculty of Science,
September, 1965.



CONTENTS

	Page
Section 1. The Need for Long-Term Planning	1
2. Historical Background	3
2.1 Definition of an "Optimal System"	3
2.2 Design of Static Optima	4
2.3 Long-Term Design Problems	9
3. The Two-stage Design Method	17
3.1 Basic Method	18
3.2 Dynamic Programming and Markovian Cost Functions	19
3.3 Next-best Designs	22
3.4 Optimality of the Solution	23
4. Selection of a Test Problem	24
4.1 Use of a Real-life Problem	24
4.2 Choice of the Problem	24
5. The First-stage Design Algorithm	28
5.1 Choice of a Design Algorithm	28
5.2 Details of the Design Algorithm	32
5.3 Major Design Changes	42
6. System Costing	44
6.1 Transition Costs	44
6.2 The Costing Algorithm	50
7. A Computer Design Study	55
7.1 Input Data	55
7.2 Results - Design Aspects	57
7.3 Results - Computational Aspects	64
8. Suggested Future Developments	72
8.1 Short-Term Developments	72
8.2 Medium-Term Developments	74
8.3 Long-Term Developments	84

	Page
Section 9. Conclusions	88
10. Acknowledgements	89
Figures 1-22	-
Appendix A1. Details of the Dynamic Programming (Shortest Path) Optimisation	A1
A1.1 Algorithm for Determination of the Optimal State Sequence	A1
A1.2 Algorithms for the Nth Best Path	A2
A2. The "Laynewspur" Problem	A7
A2.1 Simplified Representation	A7
A2.2 Cable Loading Restrictions	A12
A2.3 Further Extensions of the Method	A15
Appendix Figures A1-A9	-
Bibliography	B1
	-B5

The Long-Term Design of Electrical Power Distribution Networks

1. The Need for Long-Term Planning

"The distinguishing feature of this great soldier's mind was that it dwelt as much on the future as on the present. He was a strategist not merely in space, but in time."

- Sir Arthur Bryant describing Sir Arthur Wellesley (later Duke of Wellington).

There are several arguments in favour of any project which seeks to contribute to the state of the art of planning electrical power systems. Briefly they are:-

(a) Economics. The Economic investment in the electrical supply industry is a considerable one. (An additional £2,750 millions will have been invested in Great Britain alone over the five year period ending 1966-67.) Any savings, however small when expressed on a percentage basis, will be very worthwhile.

(b) Rate of growth. It is well-known that the demand for electrical power in industrial countries has been doubling every 10, or even 7, years. Inevitably, therefore, the electrical supply industry must plan ahead, and any discoveries which facilitate this process must help to relieve this considerable burden.

(c) The long time needed to commission plans. Even when a plan for the future has been formulated, the commissioning of the ideas it embodies may take a considerable time.

A major power station may take 5 years or more to site, design and erect, whilst even for an urban distribution substation the period may be 2 years. In unfavourable circumstances it has taken over 3 years to obtain permission to erect a pylon in a field. In view of this, the need for long-term planning is obvious.

(d) The penalty of failure. In a modern industrial society a failure to meet the demand for electric power is regarded as a serious matter. After "power-cuts" in the notorious winter of 1962-63 questions were raised in Parliament. Any techniques which can ensure the certainty of meeting future demand with a minimum of expenditure will obviously be welcome.

Since the introduction of computers (and in particular digital computers) into the field of power systems, there has been considerable development in the techniques of system analysis and static design (i.e. neglecting the variation of design parameters with time). Preliminary explorations into the field of the long-term design of electrical power systems have already been made, and the time is now right for a major attack in this direction.

Furthermore, developments and experience in the long-term design of electrical power systems would find wide application in other industrial fields.

2. Historical Background

The development of techniques for the analysis of power systems, once designed, (e.g. load flows with both d.c. and a.c. circuit representations, fault level analysis, the stability of rotating machines, the economics of generation, etc.) is so well known and documented that the only comment required here is that powerful tools for this job are readily available to the designer.

2.1 Definition of an "Optimal System"

Before proceeding further it will be convenient to define the term "optimal" system. An optimal electrical supply system may be defined as the cheapest possible system which meets a given set of load demands subject to a given set of rules or constraints. The set of constraints may be rigid (e.g. a failure of any one item in the system must not give rise to the shedding of any load), or vague (e.g. due regard must be paid to the worth of natural scenery).

A vague (or "subjective") constraint arises when it is difficult to express what is required in strictly mathematical terms. Where subjective constraints apply, it will be assumed that a number of systems optimal with respect to the rigid constraints may be evolved, and the final choice on a basis of cost v. subjective constraints left to a design engineer (see section 3.3).

The body of this report will be concerned chiefly with distribution systems only. This is further discussed in section 4.2.

2.2 Design of Static Optima

Considerable work has been done in the field of producing optimal systems to meet load demands which do not vary with time. Among the approaches which have been considered are:-

2.2.1 Trial and Error

This involves a systematic search over the whole range of possible designs. Those designs which do not satisfy the system constraints are eliminated, and the cheapest of the remainder then selected. Consider the design of a network required to connect a supply substation with 6 load substations. There are 7C_2 possible cable routes or variables (neglecting considerations of switchgear and transformer sizes). If not more than one cable will be required along any one route, each variable may take the value 0 or 1, and the total number of different designs is

$$2^{21} \approx 2 \times 10^6$$

The corresponding figure for up to two cables along each route (i.e. 0 or 1 or 2) is

$$3^{21} \approx 10,000 \times 10^6$$

These figures indicate the magnitude of the problem of optimum design. The number of calculations involved even for the small problem described would involve excessive time and expenditure, despite the advent of the "third generation" digital computer.

2.2.2 Housing Estate Design

This particular problem was considered in a paper by Grimsdale and Sinclair [1]. The starting point of the problem is the layout of a housing estate, and the loads required at every point on that estate. It can be shown that the minimum cost is incurred when the substation supplying any section of load is placed at the "centre of gravity" of that load area. The technique applied may be summarised thus:-

- (a) Select substation sites, and apportion each load to the nearest substation.
- (b) Move each substation to the "centre of gravity" of its load area, and repeat iteratively to find the final substation sites.
- (c) Calculate the theoretical cable sizes and routes.
- (d) Use a trial and error method to select the nearest standard cable sizes.
- (e) Repeat the design with substation sites moved in the direction of their largest feeders, to see if a cost saving is effected. Also possibly repeat the whole process with a different number of substations.

These techniques are obviously a combination of trial and error and theoretical methods.

2.2.3 Selection of Standard Ratings

Most power system design is done with the assumption that only equipment in a certain range of standard sizes is available. Some thought has been given to the selection of

these standard sizes by Knight [2, 3]. The approach used involves assuming an area of uniform load density, and some ideal system layout for meeting this load (e.g. a pattern of equal size hexagons with supply transformers at their centres). The total cost of meeting the load is expressed as a single complex function of those variables for which the optimum standard ratings are required e.g. transformer sizes, cable cross-sections, voltage levels, etc. The cost function is then minimised, subject to the constraint of meeting the load demand, by the method of Lagrangian Multipliers, and the corresponding values of the variables evaluated.

Whilst such an approach introduces some logic into the problem of selecting standard ratings, it is open to the criticism that load densities are never uniform, nor can ideal, symmetrical, schemes to meet them be implemented in practice. In view of this, it would be interesting to study the nature of load distributions for a large number of real cases, to see if some statistical law would be found. It might then be possible to do a large number of studies on different cases, simulated according to this law. The values of the studied variables which gave the lowest overall cost might then be determined. Extensive computing facilities would be required for this.

2.2.4 Network Design by Linear Programming

An excellent paper on this subject has been published by Knight [4]. Linear programming may be applied to any problem which can be presented in the form of a linear cost function

which has to be minimised subject to a number of linear restraint equalities or inequalities.

In the problem studied by Knight, the variables consist of the number of lines along any particular route connecting two substations. If there are n supply or load substations there are ${}_nC_2$ possible paths or variables, and the cost function is

$$C_{\text{OST}} = \sum_{i=1 \neq j}^n \sum_{j=1 \neq i}^n C_{ij} P_{ij}$$

where C_{ij} is the cost of a single circuit from substation i to substation j , and P_{ij} is the corresponding number of circuits. The approximation that the cost is directly proportional to the number of circuits can be corrected, but at the expense of a large increase in the quantity of computation required to solve the problem.

In the design of distribution networks at higher voltage levels, the dominant consideration is usually one of security of supply. This may conveniently be represented by a series of constraints to the effect that any group of t substations must have at least $q(t)$ cables supplying it. Values of $q(t)$ must be supplied for all t from 1 to n . These requirements form a number of lower-bounded inequalities. Since t substations can be selected in ${}_nC_t$ different ways, there must be

$$\sum_{t=1}^n {}_nC_t = 2^n - 1$$

of these inequalities. Further upper-bounded inequalities may also be introduced to limit the number of cables along any route, and the number of cables into any one substation. Values of the P_{ij} which give the minimum of the cost function subject to these constraints may be found by the usual methods of linear programming. The values of the P_{ij} so found will be non-integral, and some algorithm, such as that due to Gomory [5], must be used to find the nearest integral solution.

A typical problem involving 9 load substations and 2 supply substations gave rise to 55 paths or variables, and $2^9 - 1 = 511$ lower bounded inequalities. These formed a linear programming matrix of 29000 elements. Whilst the solution of a problem of this size is within the capabilities of modern computers, considerable computing time is required for it.

Various suggestions have been made for reducing the problem to more manageable proportions [6]. A general statement of the linear programming approach is given in [7], and an example of its application to another problem of system design in [8, 9].

Important features of this approach to network design are:-

- (a) It is only feasible for problems directly expressible in the form of linear cost functions and restraint equalities or inequalities. Even for the special case described above, the final solution must still be subjected to a load flow analysis. Satisfaction of the

original constraints does not guarantee the production of a network with satisfactory electrical properties.

(b) At the present time the production of the optimal integer solution from the non-integer solution presents serious difficulties.

(c) It is a technique in which the processes of design and optimisation are intermixed.

2.3 Long-Term Design Problems

For the reasons given in section 1, more and more attention is being paid to the problem of producing not merely the cheapest design to meet a fixed load pattern, but the cheapest sequence of designs to meet a load which varies with time. Evidently it is not sufficient to consider merely that sequence of which each design is the static optimum with respect to its own corresponding load demand. When the cost of conversion from one state to another is considered, a sequence of designs, none of which is optimal in the static sense, but involving the minimum of conversion costs, could well prove to be a cheaper solution.

The idea of long-term design pre-supposes that a reasonable forecast of load demand can be made. In the electrical supply industry it is common to consider a period up to about 15 years ahead. As well as load demand, the variations of interest rates and costs, and also likely technological developments must be predicted. Even if the estimates for

the distant future are not as accurate as for the near future, their effects on the choice of policy should be small. These estimates can then be revised in the light of experience when the study is repeated on a periodic basis.

Several methods have been proposed for producing the optimal sequence of designs.

2.3.1 Trial and Error

As in section 2.2.1, consider a design involving n variables, each of which may have 2 possible states. At any given time there are 2^n possible different designs. Then over a period of time divided into N discrete divisions there must be $(2^n)^N$ possible different sequences of designs. Simplifications may be effected by deleting all states which obviously will not satisfy the supply constraints. But the amount of computation involved for even the smallest of real-life problems would still be far too large to contemplate. This pinpoints the crux of the long-term design problem - the selection of one sequence of states from an extremely large number of possible combinations of alternative states.

To overcome this problem of dimensionality a simplified approach has been suggested [10-18]. This uses the basically simple idea of policy comparison. A policy is a set of rules governing the expansion of a system when conditions are such as to require extension or reinforcement. The policies used imply no form of optimisation, but usually merely present a table of alternative steps which may be tried in sequence, in any given set of conditions. By performing a number of studies

with varying policies, the cheapest overall policy may be found by direct comparison. In one typical study, this approach was used to investigate the effects of variations in policies of unit size selection, unit reliability, unit retirement, heat rate selection, etc. [15].

A slight variation on this method consists of trying all the alternative steps for each system modification [19]. In this way a "tree" of possible design sequences is built up, and each "branch" may then be costed separately to find the cheapest design sequence. If the problem is to be kept to manageable proportions, the number of alternative correcting measures, and the number of times they have to be applied, must be suitably restricted.

Whilst such "ad-hoc" approaches can not guarantee to produce that design sequence which is the true optimum, they can still provide useful background information for the planning engineer.

2.3.2 Extension of Linear Programming

The method described in section 2.2.4, for obtaining a static optimal design by the application of linear programming, may be extended to cover the problem of long-term design also. The original variables P_{ij} are replaced by P_{ijt} , (t) signifying the time interval for which P_{ij} is being considered. For each (t) there will be a different set of constraints corresponding to the loads in that time interval. Assuming that in all cases the network is extended, and that no lines are ever removed, a set of constraints of the form $P_{ij(t+1)} \geq P_{ijt}$ may be added. The cost function must be suitably extended.

The comments of section 2.2.4 are valid for this case also. In addition, the increase in the number of variables and constraints would involve a very large increase in the quantity of computation required. It is possible, however, that the decomposition algorithm of Dantzig and Wolfe could be applied to this problem. It would be interesting to see some work done in this field.

Linear programming has also been suggested for the long-term design of a complete power system [20]. However, in this case the distribution network involved was restricted to a simple radial pattern. It was further assumed that units and lines of any size were available, thus removing the necessity of finding an integer solution.

2.3.3 Sub-optimisation of Strategies

This approach is due to P. Gaussens of the Electricité de France [21]. It is based on the breaking down of a large problem into a number of smaller sub-problems, which are optimised independently. For this method a "topology" is defined as a description of the connections forming any particular design. The ratings of the individual elements, termed "dipoles", and consisting of lines, transformers, etc., are initially unspecified. The first step in the method is a series of sub-optimisations for each topology, to determine the sequence of states through which each dipole must pass.

For these sub-optimisations it is assumed that the state of any dipole may be determined independently of the states of all other dipoles in the topology. Suppose that any dipole

may have a limited number of possible states, denoted by a, b, c. Then the possible transitions which may occur are $a \rightarrow b$, $a \rightarrow c$, and $b \rightarrow c$. The optimal time t_{ab} for transition $a \rightarrow b$ is determined from the equation

$$\text{saving in running costs} = \text{expense of change-over.}$$

The other transition times are determined in the same way. If the state of the dipole is changed only at these transition times, then there exist 4 possible strategies governing the sequence of states through which the dipole may pass - Fig. 1. For a period of operation ending at a time t , the overall costs involved in each strategy may be evaluated, and the optimum strategy $S(t)$, corresponding to the cheapest cost $C(t)$, found. This is repeated for each value of t , to find $S(t)$ for the full design period considered. The combination of strategies for all the dipoles gives the optimum state sequence of the topology for a period ending at any time t .

When this has been performed for each topology, the next step is to determine the sequence of topologies giving the overall optimum design sequence. In exactly the same manner as for individual dipoles, the optimum times for transition between topologies E_1, E_2, E_3 , etc. may now be found.

$$\text{e.g. } T_{24} \quad T_{23} \quad T_{43} \quad T_{21} \quad T_{41}$$

Then possible strategies which must be considered are:-

$$E_2 E_4 E_3; \quad E_2 E_4 E_1; \quad E_2 E_3; \quad E_4 E_3; \quad E_4 E_1; \quad E_1; \quad E_2; \quad E_3; \quad E_4$$

For any period ending at some future time t , the cheapest of these may be selected to give the optimum design sequence.

Problems associated with this method are:-

- (a) Dipole states are not really independent.
- (b) The choice of topologies to be used must be specified by the designer either before or during the running of the program on the computer. This may considerably increase the time required to produce a solution. Furthermore the computer can then only find solutions within the range dictated by the experience of the designer running the program.
- (c) Even when the principle of sub-optimisation is used, it is still difficult to limit the number of different strategies which will arise from a reasonable number of topologies.

2.3.4 Parameter Optimisation by Dynamic Programming

In this method, proposed by Tsetkov [22], the system is defined as a series of parameters (C_1, C_2, \dots, C_n), optimal values of which must be found extending over the period of time $t_d - t_d - t_k$. In the final interval $t_d - t_k$ it is assumed that no change of state will take place (see section 6.1.5). Initially it is assumed that the sequences of values of all C except one, C_n say, are given. The optimal sequence of states C_n is then found by a process of dynamic programming which may be summarised as follows:-

- (a) For all possible states of C_n at t_d , denoted by C_n^d , evaluate the running costs for $t_d - t_k$ - see Fig. 2.

(b) Starting from each state of C_n^{d-1} in t_{d-1} , define S_d^k as the cost of change of state at t_d , plus the running costs up to t_k . Find the optimal value $(S_d^k)_{\min}$ for each C_n^{d-1}

$$\text{i.e. } (S_d^k)_{\min} = f(C_n^{d-1})$$

(c) Repeat (b) for t_{d-2} i.e. for each C_n^{d-2} find those C_n^{d-1} which minimise S_{d-1}^k , defined as the cost of the change in system at t_{d-1} , plus the following $(S_d^k)_{\min}$. Repeat this process until t_{q+1} is reached.

(d) The process terminates with $(S_{q+1}^k)_{\min}$ for all states of C_n at t_q . Given the initial state of C_n , $(S_q^k)_{\min}$ may be evaluated, and the optimal sequence of C_n back to t_d traced out.

Having obtained the optimal sequence of C_n , and taking this as part of the fixed data, the optimal sequence of C_{n-1} may be found in the same way. The process is continued for all C , and repeated iteratively until all sequences of C remain unchanging.

The significant points to note about this method are:-

(a) Any practical system must be represented by a large number of parameters. Since each parameter is optimised separately, and this optimisation is repeated a number of times in the course of the iteration process, a considerable amount of computation will be required to perform the dynamic programming analysis alone. Furthermore, the evaluation of every cost considered in this

process requires an analysis of the working conditions of the system being considered, which will involve yet more computation.

(b) The question arises as to whether the iterative process used will converge after a finite number of iterations. Whilst Tsetkov states "intuitively one can imagine that the process will converge quickly", he offers neither theoretical proof, nor experimental evidence to this effect.

(c) Even if this process does converge, it could well converge to some local optimal sequence. It would be necessary to change more than one parameter at a time to see if this had happened, and if necessary "escape" to the truly optimal sequence.

(d) As with the methods discussed in 2.2.4, 2.3.2, 2.3.3, this is again a method in which the processes of design and optimisation are intermixed.

3. The Two-Stage Design Method

The design of a power system must be satisfactory with respect to loading, regulation, stability, fault levels and security. The constraints introduced by each factor are often widely different in form and effect. Although only one or two factors may predominate, according to the particular system application being considered, the problem of producing even a single satisfactory design can in itself be most complex. Thus the demand for a general purpose method of producing, by a single application, a sequence of designs which is both electrically satisfactory and economically optimal, sets a difficult problem. Most of the methods mentioned in section 2 were evolved in answer to this demand.

As a compromise between the difficulties arising from the large number of variables and from the complexity of the design process, and the economies to be gained by a search for optimality, the following two-stage planning method has been evolved. The intention was to provide a method in which the processes of design and optimisation were applied consecutively rather than simultaneously. Great simplifications were then possible.

The new method uses dynamic programming, but on a basis more akin to the method of policy comparison than to the other methods reviewed in the previous section. (The theoretical details of the two-stage design method were presented by the author and his supervisor in a paper at the 1965 Power

Industry Computer Applications Conference of the I.E.E.E., which is given as reference [34] of the bibliography.)

3.1 Basic Method

Let the period over which the optimal design sequence is required be divided into intervals $t_0, t_1, \text{-----}, t_n, \text{-----} t_k$ as in Fig. 3. Each interval will have its own set of load demands and corresponding constraint equations. Suppose that for each interval t_n , a set of acceptable designs $P_{n0}, P_{n1}, P_{n \ r-1}$ may be evolved by any available means. Given the costs of converting each design to all of the permitted succeeding designs, the next step is to find the optimal sequence of designs, one being chosen from each of the intervals $t_0, \text{-----}, t_k$. For (k) intervals, with (r) acceptable designs in each interval, there are r^k possible design sequences. An exhaustive search is impractical for realistic cases.

One obvious improvement is suggested by the method of dynamic programming, originally formulated by Bellman [23]. The solution to the problem in question is particularly straightforward when tackled by this method. A brief summary of dynamic programming follows in section 3.2.1; details of its implementation for the problem in question are given in Appendix A1.1. The computational effort required is approximately proportional to $r^2 \times k$ instead of r^k .

Another approach considered was to regard the transition costs between states as being analagous to distances. The problem is then presented in the form of finding the shortest

path from the starting point (i.e. state P_{00} in t_0) to the finishing point (i.e. any of the states P_{ki} in t_k). The solution to this problem is well known [24-26]. It is particularly simple when restricted to the present form, where each interval t_0-t_k must be visited once only, and in a particular sequence. In fact it turns out that in terms of actual arithmetic operations the two methods are identical.

In either case the processes of design and optimisation have now been completely separated. The second stage analysis is mathematically stable, and entirely independent of the first stage design process. Thus changes to the design process may be made freely. Additional constraints applied at the design stage may actually speed up the process of computation by limiting the number of designs to be investigated. The basic flow diagram for the two-stage method is given in Fig. 4.

3.2 Dynamic Programming and Markovian Cost Functions

The general principles of dynamic programming are now so well known that a detailed description is not required here. A brief summary will be given, however, in order to draw attention to a possible source of difficulty when the method is applied to the problem under study.

3.2.1 Dynamic Programming

Dynamic programming is a theory applicable to any problem which may be presented in the form of a multi-stage decision

process. For the purposes of the problem in question it is only necessary to consider discrete time, deterministic, multi-stage decision processes. Suppose a system may be characterised by a state vector

$$X(i) \equiv (x_1(i), x_2(i) \text{ -----}, x_n(i))$$

where (i) denotes the discrete time interval. Further suppose that at each stage in (i) a decision $U(i)$ has to be made, resulting in the transformation

$$X(i+1) = G[X(i), U(i), i]$$

Dynamic programming is concerned with the maximisation (or minimisation) of some cost function

$$I = \sum_{i=1}^N H[X(i), U(i), i]$$

subject to given restrictions on the $U(i)$.

The solution is obtained by appeal to the "Principle of Optimality" which states that an optimal policy has the property that, whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the initial decision. From this principle the well known functional recurrence equation of dynamic programming may be developed:-

$$f_N(C) = \max_{U(1)} \left\{ H[C, U(1)] + f_{N-1}(G[C, U(1)]) \right\}$$

where $f_N[C]$ is defined as the value of I obtained by using an optimal policy over N stages starting from $X(1) = C$. (This is a simplified form of the functional recurrence equation,

applicable to stationary processes, i.e. processes in which G and H are independent of (i) .)

3.2.2 Markovian Cost Functions

It is evident that the problem under study (viz. the selection of one state from each of the intervals t_0, \dots, t_k such that the overall systems costs are minimised) is suitable for treatment by dynamic programming. It is important to note, however, that in the above statement of dynamic programming, functions G and H must be independent of $X(i-1)$, $X(i-2)$ etc., or in other words, of the history of the system. This is sometimes restated in the form that the cost function to be analysed must exhibit the "Markovian" property that the optimal future policy must be determined only by the starting state of the system, and any future decisions which are made.

If this requirement is not satisfied, then the Principle of Optimality is invalidated, and dynamic programming no longer applicable. When applied to distribution system design this means that the costs to be analysed must be independent of equipment age. It will be shown that this condition can be satisfied, but some care is necessary in selecting the methods of capitalisation. A detailed discussion of this matter is given in section 6.1.

These requirements are really another way of stating that the state vector $X(i)$ must include all the relevant information concerning the system under study. Only when $X(i)$ contains insufficient information about the system is

it necessary to investigate the history of the system in order to find that information. Thus if the distribution system costs were dependent on equipment age, it would still be possible to use dynamic programming if the equipment age was included in the state vector describing a system.

Unfortunately this would have the effect of greatly increasing the number of states to be analysed, e.g. two systems made up of identical equipment, but with one transformer of a different age in each case, would have to be described as being in two different states. Thus the size of the problem would be so greatly magnified that there would be little gain compared with the method of exhaustive search.

Shortest path theory is in essence a special case of dynamic programming. The distances which make up the cost function must always exhibit the required Markovian property.

3.3 Next-Best Designs

A design engineer requires more information than a bare description of the best sequence of designs. In order to arrive at the best design sequence it is necessary to assume the values of certain design parameters (e.g. load forecasts, interest rates, etc.). The engineer will want to know what effect variations in these parameters will have upon the implementation of his design sequence. There may be other design sequences which, though not optimal under the preferred parameter values, would cost less to modify in the event of variations in these parameters. The value to be attached to

stability under parameter variations is frequently too subjective to be expressed mathematically. Other aspects of electrical networks also involve subjective judgements (e.g. the importance of preserving natural features and scenery).

In the evaluation of such subjective matters, the 2nd, 3rd ----, nth next best design sequences are also of interest. Both dynamic programming [27] and shortest path theory [26, 28] provide a means of determining these sequences. In the case of dynamic programming the increase in computation is directly proportional to the number of sequences required. But for the shortest path problem, advantage may be taken of the computations already performed to find the best sequence, and the succeeding sequences may be determined with comparatively little extra labour. Details of the actual method used are given in Appendix A1.2.

3.4 Optimality of the Solution

The two-stage design method can not guarantee to find the best of all possible design sequences unless every possible design state for each design interval is included. In practical cases an extremely large number of acceptable states would exist for each interval. But fortunately experience and common sense frequently suggest that very many of these states are obviously non optimal. It is then possible to work within a restricted range of states. The uncertainty of finding the truly optimal solution must be weighed against the power and cost of the available computing facilities.

4. Selection of a Test Problem

4.1 Use of a Real-Life Problem

The flow chart of the two-stage approach to long-term design has been given in Fig. 4. To test the usefulness of this approach it was decided to choose a practical problem in the field of power system design which could act as a test case.

The study of a real system introduces certain complexities. It is no longer possible to choose idealised situations or to make broad assumptions in order to shape a problem capable of neat and straightforward solution. Furthermore, a particular problem is likely to involve difficulties peculiar to itself, which may possibly detract from the general conceptions under investigation. However, the whole aim of the project was to provide a method of use in real cases. Also, a specific claim of section 3 was that the approach suggested should be of use whatever the nature of the problems involved in producing feasible designs. Thus it was felt that the only fair test would involve the study of a real-life system.

4.2 Choice of the Problem

Since all parts of the British power system are inter-related, even if only very indirectly, any attempt at optimisation should theoretically involve the whole system. (In the presence of the cross-channel d.c. link the continental power system should also be included.) Such a problem is far too large to contemplate, and at the present time only some very

small portion of it may be tackled. Fortunately the constraints of geography and economics (which require the use of several distinct voltage levels within the system) produce divisions enabling one to isolate parts of the system, which are at least to some extent independent.

It was felt that it would be advantageous to study a problem which arose in an area local to the author's university. The South of Scotland Electricity Board, and in particular the local Edinburgh district board, were most willing to co-operate on a design study. Therefore, it was decided to undertake an investigation of the problems involved in the long-term design of the power system of the Edinburgh district. The advantages of studying this easily accessible problem were two-fold. Firstly, expert advice on the technical problems arising in the design process was readily available (and freely given). Secondly, comprehensive records of the past history of the Edinburgh system were available. It was felt that these would provide a useful basis for a retrospective study, and a good yardstick by which to measure the results.

Even when the area of study had been restricted to that of the Edinburgh system, it was felt that initially an even smaller problem was required. The Edinburgh system may be divided into 4 distinct levels, distinguished primarily by their working voltages:-

- (1) Supply of power, from the National Grid at 275 (or 132) kV to 33 kV substations, and from local generation at 33 kV.

- (2) Distribution of power at 33 kV to 11 (or 6.6) kV substations.
- (3) Distribution of power at 11 (6.6) kV for transformation to the voltage required by individual users.
- (4) Supply of individual users.

Level two, that of distribution of power at 33 kV, was chosen as that most suitable for an initial investigation. This was because it held sufficient scope for development over a period of time, whilst not involving a large mass of detail.

Obviously the part of the system at 33 kV could not be treated as completely independent, either of the supply system at higher voltage, or of the load system at lower voltage. But the areas of interaction were sufficiently confined to enable a system of "parameterisation" to be used. It was assumed that the states of the higher and lower voltage systems could be taken as fixed data during the design of the 33 kV system. The inter-relationship between the various voltage levels could then be investigated only by repeating the design process with different sets of data. It was decided that only after experience had been gained with work on a subsystem at a single voltage level should an attempt be made to cover more completely the problem of a system with more than one voltage level.

4.2.1 Specification of the Problem

The aim of the initial study was therefore to use the two-stage design method to study the long-term design problems of the 33 kV Edinburgh distribution system. It was assumed that the following information would be supplied as data:-

(a) Loads on the 33/11 kV substations over a period of years. In general several substations were grouped together to supply a single independent "area" or "block" of 11 kV load. Because of the close electrical inter-connection, it could be assumed to a good degree of approximation that the 33/11 kV transformers shared the area load in proportion to their admittances. Thus the load data could be given as the grouping of substations into "areas", and the corresponding total load in each area. The physical locations of the load substations were also required.

(b) Locations of the 275 (132) kV supply substations, and possibly also the maximum supply which could be taken from each.

4.2.2 Use of a Computer

It was obvious that the help of a digital computer would be essential for this problem. In view of the complexities of distribution system design, it seemed reasonable to assume that a very powerful computer would be required. Details of the "Atlas" computer actually used are given in section 7.3.1.

5. The First-Stage Design Algorithm

5.1 Choice of a Design Algorithm

Having specified the area of study, the next step was to select a suitable design algorithm to formulate the sets of possible designs for each of the design intervals - i.e. for stage 1 of Fig. 4. It was appreciated that with the particular problem chosen, as indeed with any real-life problems, the major portion of the project's programming effort would be required on this part of the study.

One possibility considered was to try to make use of some form of algorithm which produced a static optimal design (e.g. as in section 2.2.4). If a static optimal design for any interval was obtained, it could possibly be used to evolve permutations of different non-optimal, but otherwise acceptable, designs. Or conversely, if the optimal design was obtained via a sequence of non-optimal, but otherwise acceptable designs, (as is the case with linear programming), then this sequence could form the group of possible designs for the corresponding interval. Unfortunately there seemed to be no method of obtaining a suitable algorithm for producing even a static optimum for the specific problem under consideration.

5.1.1 Ad-hoc Design

It is a feature of the design of distribution networks that rarely is it required to design a whole new system at any one instant. Usually, in any one design interval, it is only required to add a few new substations to what is already an extensive network. Furthermore, very often the load growth

is such that in a new interval only certain parts of the network tend to become overloaded. The problem is not then one of complete redesign, but rather of selecting one of several, frequently obvious, alternatives for extension of parts of the network. Indeed, it is for this reason that design engineers can cope at all with the problems of long-term design.

Hence it was decided that an "ad-hoc" or common-sense algorithm, based on the logic used by a design engineer, might be successfully incorporated. The same conclusion might well be reached in many problems of real-life design, where no more attractive alternative is available.

5.1.2 Implementation of Ad-hoc Design

The writing of an ad-hoc design program may itself be a difficult project. The design algorithm is required to generate all the designs which it is considered would make likely candidates for the optimal design sequences. But at least it does not have to be so good that it produces only these designs. If some designs are produced, which a design engineer would discard as obviously non-optimal, these will be rejected automatically by the dynamic programming analysis. The only loss in this case is the waste of computational effort.

It is impossible to foresee every design situation when writing the design program. There may well be cases where the design program is unable to modify a design for the next interval. The whole program run is not invalidated however. The program may continue to operate, though the only designs

in the next interval will be those derived from the other designs which could be modified. If some indication of these events is given in the results print-out of the program, steps may be taken later to extend the design algorithm to meet the new situation. Meanwhile some useful results will still have been obtained.

5.1.3 Restricting the Number of Designs

The concept of "obvious alternatives", mentioned above, implies that the designer's experience is being used to restrict the range of solutions investigated. But since each design in one interval can give rise to several alternative designs in the following interval, it was thought that some further method of confining the growth of alternatives might still be required.

Although the design changes from interval to interval are often of a local nature, this is not always true. Occasionally a radical reorganisation of the old system is called for. This is frequently due to some change in an external system. For example, in the actual problem chosen for study, when all the supply substations are loaded to capacity, another substation must be introduced, and the 33 kV system reorganised to re-distribute the supply substation loads. Such a time of major reorganisation could be a convenient time to restrict the number of completely new systems which are considered.

Designs produced in the early design intervals may not be discarded, since they are required for the subsequent dynamic programming analysis. If the size of high-speed

computer store is limited, the problem may be eased by dumping the early designs on to magnetic tape (or in some other form of low speed store) until required for the final analysis.

However, it was felt that the speed and size of the "Atlas" computer were such that, at least in the early studies, it would not be necessary to implement either of these measures.

5.1.4 Relationship to Other Design Methods

Although simulating the design logic of the engineer, a two-stage design approach based on ad-hoc programming differs greatly in one respect from the usual approach of a design engineer. At each interval the design engineer usually chooses only one of the best alternatives as preferable to the others, and continues with this one design (or at best 2 or 3) to the next interval. But the two-stage algorithm carries all alternatives forward to the next interval, and delays the choice of alternatives until designs for the whole set of intervals have been formulated.

The production of alternatives is similar to the method in which a "tree" of possible sequences is completed, and each "branch" of the tree costed to find the best sequence (section 2.3.1). The two-stage method differs from this in allowing design sequences to change from one branch to another in any interval. It is similarly more extensive than the method which uses different policies to generate different design sequences, in that changes from one policy to another are also considered at all intervals in the design period.

5.2 Details of the Design Algorithm

Investigation showed that the basic design problem could be split into three sections, each being to some extent independent of the others. These sections were:-

- (1) Arranging supplies to new load substations as demanded by the problem data.
- (2) Checking the loads on transformers in both normal and fault conditions, making provision for an increase in the size of one or more transformers if required.
- (3) Checking the loads on cables in both normal and fault conditions, making provision for the alleviation of overloads if necessary.

The design algorithm was based on these three sections. Obviously if a number of alternative network modifications were to be allowed, each section could give rise to some alternatives. For the sake of simplicity the initial algorithm was written such that only section (1) would put forward a number of alternative plans. For sections (2) and (3) it was decided that if modification of a system was required, then only one possible modification would be allowed (see section 5.2.4).

The basic flow chart is shown in Figs. 5(a) and (b). The algorithm consisted of three basic program loops. The outer loop was performed once for each interval within the design period. The next loop was the actual design portion of the algorithm, which was performed once for each possible starting system within the design interval. Within the first part of this loop a number of alternative new systems were

generated to include supplies to new load substations. This corresponds to section (1) above, and was performed by the routine named "menews" (meet new supplies). The inner loop of the design algorithm was then performed once for each of these new systems, and consisted of sections (2) and (3) above, and named respectively "trach" (transformer check), and "uloch" (unit load check).

The writing of the algorithm was complicated by the history of the Edinburgh network. The early network was based on the use of closed rings of interconnected substations. Each substation employed its own circuit-breaker which could be used to break the ring in the event of a fault. (This will afterwards be referred to as the "ring-main" system.) At a later date this policy was dropped in favour of the use of open ended spurs. Each spur could consist of one or more substations connected in series, with a circuit-breaker at the supply end of the spur only. (This will be referred to as the "spur" system.)

In order to perform a retrospective study of the Edinburgh system, it was hence necessary to have a design algorithm which could cope with either situation. Thus in reality two design algorithms were used, one for each policy. Initially it was assumed that only one policy would be used for each interval, the choice being controlled by the input data.

For a complete study of the relative merits of each policy, both should be used simultaneously for each interval. This is particularly so if it is required to determine the optimal

date of change from one policy to another. However, this extended use of the design algorithm was left for later development.

Details of the three basic sections of the design algorithm are given in sections 5.2.4, .5, and .6.

5.2.1 Storage of the System Data Within the Computer

"System" is used to indicate a complete distribution system which satisfies the load supply criteria for any given design interval. A "unit" is defined as a group of interconnected load transformers, and the cables, switch-gear, etc. associated with them. Thus a system is composed of a set of units, which may be of ring-main or spur pattern, or of both.

The basic item of storage was the unit. Each unit was identified by a number denoting the first location in the computer store where information concerning that unit was kept. Each item of information describing a unit could be addressed by the use of a name and some identifying suffices. For example, "unno(u)" indicated the number of transformers in unit number (u). A special subroutine (known as a "mapping function") was written for each name, so that the corresponding location in the computer store could be determined whenever a name and its associated suffices appeared in the program. As the program was developed, it was necessary to re-arrange the storage pattern of unit information a number of times. When this occurred it was only necessary to change the routines for translating the names, and not to change every reference to the names within the program. Details of the information

actually stored for each unit are given in Figs. A3(a) and (b) of the appendix.

Each system was then stored as a list of unit numbers. The lists corresponding to each system were stored consecutively in what was known as the unit list. In order to locate the portion of this list corresponding to any system, a system list was also formed. There was one entry for each system, indicating the start of the corresponding portion of the unit list vector. The end of the list for each system could be found immediately as the location preceeding the start of the list of units of the following system. The systems of a particular design interval were grouped together, and markers to denote the end of one interval and the start of another placed at the corresponding points in the system list.

The hierarchy of lists may be summarised as

system list → unit list → individual unit information lists

5.2.2 Geographical Mapping of Systems

Since the design algorithm had to deal with distribution systems situated in a city, a basic constraint was that all cables could only be laid along roads, and not along arbitrary straight lines. Thus a means was required of representing the permitted road plan within the computer. All road junctions - termed "nodes" - were numbered. Each road was then represented as the pair of node numbers corresponding to its end points, and a number denoting its length.

It was assumed that cables would always be laid along the shortest possible road route. A subroutine was written to derive the shortest route between any two specified nodes.

The method used is described in [25].

5.2.3 Correlation of Unit and Load Area Data

In dealing with the problem of system modification it was sometimes preferable to deal with the system in a unit by unit manner, and sometimes in a load area by load area manner. In order to change from one mode of operation to the other, 6 sets of lists (a set comprising one list for each load area) were set up for each system as it was modified. These were:-

Set No.	Name	Information
(1)	arnode	Nodes making up corresponding load areas.
(2)	usa	Identification number of the unit supplying each node.
(3)	usar	Corresponding rating symbol of the unit transformer.
(4)	Taln	Normal load on the transformer.
(5)	Taff	Factor by which all other transformer loads in the same area would be increased if this transformer was removed due to a fault.
(6)	toss	Indication of which transformers (being on the same spur) could lose their supply simultaneously.

Set (1) was assembled directly from the input data, and was fixed for any design interval. Sets (2) and (3) were assembled by the subroutine "dulos" (see Fig. 5), and sets (4), (5) and (6) by the subroutine "tralaff". The reason for using two separate routines will become evident in the next section.

5.2.4 Details of Basic Routine "menews"

New load transformers could be supplied either by extending existing units to include the new transformers, or by laying down completely new units specifically to supply them. In either case the possibilities were conditioned far more by the geographical relationships between the existing system and the new transformer positions than by considerations of load flows within the existing system. The consideration of various possible geographical patterns gave rise in general to a large number of alternative solutions. This was in contrast to the problems arising when a transformer or cable overload required correction, and the number of obvious or common-sense alternative solutions was more limited. It was for this reason that the basic routine "menews" was selected as the one to put forward a number of alternative new systems, whilst the remaining two sections were originally allowed only one possible solution to each problem. It was these considerations which largely dictated the form of the basic flow chart of Fig. 5.

Since the problem of meeting new loads was to be solved using only geographical considerations, only list sets (2) and (3) of the previous section were required for the first part of the algorithm. Sets (4), (5) and (6), concerned with load flows within the units, were only required for the latter part of the algorithm. This is why two separate routines were used for assembling these lists. Sets (2) and (3) had to be re-computed after the action of "menews" since unit numbers

were changed whenever units were modified in any way.

The basic flow charts for the two parts of "menews" dealing with spur and ring-main designs are given in Figs. 6 and 7 respectively. They are similar in general outline. In each case the first step was to group together new load points according to their geographically nearest existing units. For each group two alternative ways of providing supplies were allowed. Thus, if there were N groups of new load points, then 2^N different modified systems would be produced.

In both cases the first alternative was to modify the unit considered to include all the new load points nearest to it in the simplest possible manner (i.e. using the least amount of cable). In the case of spur designs the possibility of the nearest unit being either of spur type, or of ring-main type from some earlier date, had to be provided for.

For the second alternative with spur designs a further sub-division of each group was carried out, so that the load points of each subgroup had the same nearest 275 (132) kV supply point. Each sub-group was then arranged into as many new spur units as were necessary to meet all the load points from the supply point using the minimum amount of cable. The subroutine used to do this, entitled "laynewspur", is discussed in appendix A2.

Unlike the equivalent case involving spur designs, there were seldom enough new load points inserted in any one design interval to allow a second alternative of a completely new ring-main unit specifically to supply them. The alternative

in this case was to again include the new load points in their nearest existing units, but at the same time to reinforce these units. This was achieved by laying an additional cable from the grid supply point to one of the load points already on the ring of the ring-main. This reinforcing cable was called a "radial feeder", and was also used to supply some of the new load points.

If any of the new points lay geographically within the ring, then the radial feeder was used to supply them, the load points being ordered along the feeder according to distance from the supply point. If no new load points lay within the ring, then a selection of the points to be included on the feeder was made with the aid of the "laynewspur" subroutine. This subroutine was used to find the spur pattern which would connect the new load points to the supply point with least cable, ignoring the presence of the existing unit. The spur with the most load points on it was then converted into a radial feeder by connecting its free end to the nearest load point already on the unit.

5.2.5 Details of Basic Routine "trach"

The flow chart of "trach" is given in Fig. 8. This routine performed a check area by area on transformer loads, firstly under normal conditions, secondly under fault conditions. The actual check in each instance was performed by the subroutine "choltrina" (check loads of transformers in each area), the flow chart of which is given in Fig. 9.

Under normal conditions the normal transformer load was compared directly with its nominal rating. For fault conditions the transformer with the largest fault factor was failed, and the other transformer loads increased accordingly. The transformer ratings were also increased from nominal to their short-time ratings.

In the event of an overload, only one way of modifying the existing system was to be allowed, as discussed previously. The modification was performed by replacing one transformer by another of next larger rating. The replaced transformer was to be one of those with the minimum fault factor for the area. The actual transformer was chosen such that, with the larger transformer in place, the corresponding maximum cable load in its own unit was a minimum compared with equivalent maximum cable loads obtained by increasing the sizes of the other transformers. Thus the probability of the need for later cable reinforcement was reduced to a minimum.

5.2.6 Details of Basic Routine "uloch"

The flow chart of "uloch" is given in Fig. 10. The basic loop of this routine was performed once for each unit in the system. A load flow study was first performed with normal loads on all transformers. The study was then repeated with the increased transformer loadings which could arise from the failure of transformers on other units which supplied any area in common with the unit under examination. These conditions could arise from:-

- (1) The failure of a whole spur unit.
- (2) The failure of a single transformer on a ring-main (only the largest in any area being considered.)

Once again, in the event of an overload, only one course of correction was allowed by this routine. If the overloaded unit was of spur type, the action taken was to split the unit into two new spurs. The split was made just a sufficient number of load points along the spur from the supply point to alleviate the overload - thus using a minimum length of new cable. If the overloaded unit was of ring-main type then radial reinforcement feeders from the supply substation to each load point on the ring-main were tried in turn. A list of all feeders which alleviated the overloads was compiled and stored for use later in the routine.

At this stage all checks on a spur unit were complete. It then remained to check the load flows in the ring-main units, with each cable in the ring-main loop failed in turn. In these cases the transformer loads were taken at their normal values, possibly modified by the loss of the transformers at either end of the failed cable, as determined by the switch-gear layout of the unit. The correction of overloads was again by the use of radial feeders. If an overload had occurred during the check of load flows with transformers faulted, then only those feeders which had also been found to alleviate the earlier overloads were considered. The radial feeder which reduced the maximum cable load on the unit to a minimum was then used to reinforce the unit.

5.2.7 Note on Load Flow Studies

Load flow studies on spur units were of a trivial nature. Load flow studies on ring-main units were facilitated by the use of the admittance matrix and its inverse, which were stored together with the other unit information (see Fig. A3 of the appendix). It was assumed that sufficient accuracy could be obtained by a d.c. representation of the various impedance values - i.e. each impedance was treated as if it were a simple resistance of the same magnitude. (This was the assumption usually made by a design engineer. For a discussion of the errors involved see [29].) Loads were assumed to have a constant current characteristic, and the iterations required for constant power studies were thus avoided.

Studies involving the loss of an existing cable or the insertion of a new one were conveniently performed by the application of a correction to the old inverse matrix. (Methods of inverse correction are discussed in [30].)

5.3 Major Design Changes

The idea of major design changes, involving the introduction of new 275 kV supply points and a complete reorganisation of the 33 kV system, was mentioned in section 5.1.3. The design algorithm as it has been described above was not capable of dealing with such a design situation. This was because the level of design experience and common-sense required to deal with such a situation was of a degree higher order than that required for the minor design changes. Since a considerable programming effort had already been required on

the minor order design problems, it was thought to be more important to get this portion actually working than to continue deeper studies into the area of major design changes.

Furthermore, it was felt that since major system changes were linked with changes to the 275 kV system, these problems were better dealt with in the context of the design of multi-voltage level systems, which had already been set aside for later study.

6. System Costing

6.1 Transition Costs

A transition cost is defined as the sum of the capital costs incurred in changing a system from one state in a given design interval into another state in the succeeding interval, plus the costs involved in running the new system for the whole of the new interval. In order to compare the relative merits of incurring different costs at different times, all such costs must be corrected to their corresponding values at some similar moment in time. In this study the common conception of "present worth" was used - i.e. the time of study was taken as the basic reference time.

As stated in section 3.2, for the dynamic programming (or shortest path) analysis to be valid, the costs to be analysed must show certain "Markovian" properties. To see if this is so for electrical distribution systems, their separate cost components must be analysed in detail. These components are:-

A. Running Costs

1. Power losses
2. Maintenance costs

B. Capital Costs

1. Installation and removal costs incurred at system changeover
2. Capital charges on all equipment used by the new operating system
3. Capital losses on equipment no longer used in the new system -

- (a) Abandoned equipment of no further use
- (b) Equipment which may be re-used elsewhere

Of these, the items listed under A usually exhibit the required Markovian property. Whilst to some extent maintenance costs may vary with equipment age, they are frequently so difficult to determine accurately that they are budgeted for merely as a fixed percentage of the corresponding new capital costs. Capital costs B(1) will certainly also exhibit the required property, but items B(2) and B(3) require closer study.

6.1.1 Methods of Capitalisation

There are numerous approaches to the capitalisation of engineering works. Suppose some piece of equipment is purchased at an initial cost V_0 [21]. Further suppose that it has a lifetime of T years, and that at the end of this time its residual value is V_T . One method of payment would be to lay down immediately the full sum V_0 . At the end of T years a sum of V_T may be recovered. Hence the total effective cost referred to the time of installation is given by

$$C_{OST} = V_0 - \frac{V_T}{(1+i)^T}$$

where i is the current interest rate.

A second method assumes that initially V_0 is borrowed. Then in each year t , a sum

$$a_t = iV_0 + r_t$$

is set aside (into what is commonly called a "sinking fund", where iV_0 provides the interest on the borrowed capital, and

r_t provides for the depreciation in the worth of the equipment from V_{t-1} in year $(t-1)$ to V_t in year t . If the total cost referred to the year of purchase is to be the same for both methods of capitalisation, then

$$\sum_{t=1}^T \frac{a_t}{(1+i)^t} = V_0 - \frac{V_T}{(1+i)^T}$$

The quantities a_t , r_t and V_t are not independent. It is difficult to determine the value V_t for any equipment, owing to the lack of a second-hand market. In the absence of any other information, and for the sake of convenience in accounting, it is commonly specified that a_t , and hence r_t , should remain constant. Thus

$$a_t = a = \frac{i}{(1+i)^T - 1} (V_0(1+i)^T - V_T)$$

Either of these costing methods meets the requirements of the study for item B(2), namely that any cost incurred should be independent of the age of the equipment, and hence of the previous history of the system.

6.1.2 Allowance for redundant Equipment

The effects of these two charging methods on the capital losses, B(3), on equipment no longer used in a new system have still to be studied. Such equipment may be either (a) abandoned, or (b) re-used elsewhere.

Consider the first method of costing (a single initial payment). Type (b) equipment which is removed at age t will be sold, providing an income of V_t - Fig. 11. The evaluation of V_t would involve the age of the equipment. But for equipment

of type (a) no income has to be evaluated; the equipment merely stands idle.

Now consider the second method of costing (equal annual instalments). When type (b) equipment of age t is removed, the only point to be considered is that all payments cease. The equipment may be sold for just that sum which, when added to the accumulated investments of a_t , repays the outstanding load². When type (a) equipment of age t is removed, two courses are possible. One is to pay back the loan immediately, incurring a loss of V_t - a function of the age of the equipment. The other course is to continue the annual instalments over the full T years. But the actual date of the final year is again dependent on the age of the equipment.

6.1.3. Choice of Costing Methods

Thus in order to avoid methods of costing which involve computation of quantities which are a function of equipment age, it is necessary to use method (1) with type (a) equipment and method (2) with type (b) equipment - as indicated in Fig. 11. The simultaneous use of two costing methods is merely a convenience to avoid using the age of the equipment.

In this study, transformers and switchgear were taken to be of type (b). The distribution lines took the form of underground cables, and were taken to be of type (a).

With this in mind, the various component costs may be regrouped under the two headings of operation costs and conversion costs. The operation costs include all the items of recurrent annual costs. These consist of the running costs,

listed under A above, and the annual capital charges on type (b) equipment. The conversion costs are those costs which are incurred once only, when a conversion is made from one state to another. The installation costs of B(1) are included in this group. To them are added the total costs of any new type (a) equipment.

In a really long-term study the possibility of having to replace worn out equipment must be allowed for. Using method (1) for costing, a further payment of V_0 would then be required for either type of equipment. The actual date of payment would depend on the date of installation, again a function of system history. However, since most electrical equipment is designed for a life of 20 years or more, and most planning studies consider a period of the order of 15-20 years at the longest, this factor may usually be neglected. But in any case its effects will be reduced by the present worth calculation.

6.1.4 Utilisation of Redundant Equipment

A careful choice of accounting methods would thus appear to eliminate the effects of the non-Markovian property of costs involved in making equipment redundant. But unfortunately this factor still affects the problem solution. During both the design and costing stages no attempt can be made to make use of type (a) equipment which has been abandoned in some earlier interval of the design period. At the design stage it is not apparent which abandoned equipment (if any) will be available, since it is not clear which states from previous

intervals will be chosen to precede the one under examination. Even if this were possible, then which equipment was free for use would be a function of the previous history of the system, and so would be the transition costs, thus invalidating the dynamic programming analysis.

The first analysis of results must be made ignoring this factor. Once the cheapest sequences have been evaluated, an examination may be made to see if any further improvements can be made by taking advantage of redundant type (a) equipment.

However, these restrictions do not apply to equipment with age-dependent costs which is found in the starting state (i.e. P_{00} in t_0). In any succeeding state, this equipment will always have the same history (i.e. started in t_0), and the dynamic programming will not be invalidated.

6.1.5 Terminating the Design Period

Another point to be noted is the relative importance of operation and conversion costs during the final interval of the design period. Since in the proposed costing method the full capital costs of type (a) equipment are included in the conversion costs, the figure for conversion costs could considerably exceed the figure for annual operation costs. If the final interval was designated as a single year, therefore, the economic analysis would tend to favour state sequences ending in lower conversion costs and higher operation costs. It is true that a designers interest will be focussed on the early intervals in the design sequences, and also that the effects of the events at the end of the design period are

greatly reduced by the present worth valuations. But the disproportion between operation and conversion costs should still be corrected if possible.

It is evident that at the end of the design period the whole system will not be abandoned, but will continue in operation, though modified by design changes not considered in the study. To reflect this continuation, and preserve a balance between operation costs and conversion costs, the last interval in the design period should be set to a length corresponding approximately to the average equipment life. (See also [22].)

6.2 The Costing Algorithm

The flow-chart of the cost analysis routine is given in Fig. 12. The list of systems in each interval (described in section 5.2.1) was first scanned to assess the amount of computer store which would be required both for the information concerning transition costs, and for working space during the dynamic programming analysis.

The complete set of transition costs was then evaluated. Each cost was formed as the sum of the operation costs incurred over the whole of the design interval, plus the corresponding conversion cost.

Since the dynamic programming analysis was more conveniently written to deal with a problem involving a single possible end state, a dummy end state was introduced, together with the corresponding transition costs (all zero). The routine for analysing the cheapest design sequences was then entered.

6.2.1 The Transition Cost Routine

The basic flow chart for this routine is given in Fig. 13. A list of the units in the new system was first compiled. Part of the information stored with each unit was the figure for the annual operating costs. The operating costs of the whole new system could hence be calculated. A list of the units in the old system was then compiled, and the units common to both lists deleted, as these would obviously involve no conversion costs.

To compute the conversion costs of removing old transformers and installing new ones, lists of transformers in the remaining units of both the old and new systems were compiled. Transformers appearing in both lists with similar ratings and installed in similar locations were deleted. The removal costs of the remaining transformers from the old system, and the installation costs of the remaining transformers from the new system, were then added to the total conversion cost. Costing of the removal and installation of switchgear was performed in a similar manner.

Only small variations of this approach were required to cost the laying of new cables. There was no question of removing old cables which did not appear in the new system. The routes of new cables had to be broken down into the individual sections between pairs of map nodes, since a fixed laying charge was applied for a given route however many cables were involved.

6.2.2 The Dynamic Programming Analysis Routine

Theoretical details of the dynamic programming analysis



are given in appendix A1. The basic flow-chart for this routine is given in Fig. 14(a).

Figure 14 (b) shows another version of the same flow-chart, this time produced on the "Atlas" computer. To speed the documentation of the whole long-term design program, a special program was developed which would accept as data any autocode program (on 7-track paper tape), and output the same program instructions rearranged in a flow-chart format. Figure 14(b) is a typical example of the output from the "flow-chart" program. It also illustrates the nature of the "Atlas Autocode" language in which the whole design program was written.

The first part of the dynamic programming analysis routine was concerned with finding the best (i.e. cheapest) design sequence. The term "expense" is used in the diagram to denote the minimum cost involved in arriving at a design state via any sequence of earlier states.

The expenses of the starting states were first set to zero. The expenses involved in arriving at each state in successive intervals were then computed.

Each expense was determined as the minimum of the sums of the expense to reach each state in the previous interval plus the corresponding transition cost between the two states. The expense calculated for the single state in the final interval, (i.e. the dummy state inserted by the general cost analysis routine), gave the minimum overall cost. The sequence of states giving rise to this cost could then be traced backwards from the end state to the starting state.

If only the best design sequence was required, the analysis was complete, otherwise the second part of the routine proceeded to find the next-best design sequences. For each previous next-best sequence, the extra costs of deviations from that sequence were computed. Each new deviation could possibly form some later next-best sequence. Deviations from the whole sequence were not investigated, but only those from the initial part of the sequence, as far as, but not including, that transition which itself formed the deviation from an earlier sequence (see Appendix A1.2.2).

Whilst the lower order next-best sequences were being formed, it was possible that the number of deviations so far investigated had not yet exceeded the total number of next-best sequences which was required. In this event the new deviations were immediately stored as possible candidates for later next-best sequences.

If, however, sufficient candidates for all later sequences had already been stored, a test was made to see if the new deviation involved less extra cost than any of the earlier candidates. If not, it was discarded. Otherwise it was entered in the list of candidates in place of that candidate which had previously involved the maximum extra cost.

When all the new deviations from the previous next-best sequence had been investigated, the deviation from the current list of candidates which involved the minimum extra cost was selected as the basis of the next of the next-best sequences. Thus each successive next-best design sequence could be computed.

Finally all the results so obtained could be output from the computer.

7. A Computer Design Study

Since good records of the Edinburgh distribution system were available, it was decided to perform a retrospective study, making use of the load figures actually monitored on the system. For the preliminary study attention was confined to a small portion of the 33 kV system over a comparatively short time scale, in order to gain experience in the use of the program.

A portion of the north-west Edinburgh system was chosen as being relatively independent. The study was to start with the system as it had developed by 1955, immediately after a major reorganisation of the 33 kV system and the introduction of a new 275 kV supply point. The next major reorganisation of the 33 kV system for this part of Edinburgh did not take place until 1963/64, thus leaving a period of 7 years over which the design program could be directly applied. In fact the first design study was performed for the first 6 years of this period.

7.1 Input Data

The input data for the study may be briefly summarised as follows:-

1. Edinburgh map data. The road network of north-west Edinburgh was represented by 161 road junctions or "nodes". Details were supplied of the approximate (x, y) co-ordinates of each node, and of the 249 connecting roads or "links" (see section 5.2.2). A list was also given

of which "nodes" formed the external boundary of the map, as this information was needed to compute whether a given point was situated inside or outside of a closed ring of cable.

2. Grid supply points. The locations of the 275 kV grid supply points were specified (one only in the first design study).

3. Basic cost data. The interest rate on borrowed capital (6%) and the expected lifetime of equipment (20 years for all equipment).

4. Transformer and switchgear data. This included technical data and cost data for the individual items of equipment. The details are given in Fig. 15. (The cost data was of an approximate nature only, of sufficient accuracy to illustrate the validity of the design method, but not for use in real-life design comparisons.) Since only one size of cable was used, its technical and cost details were permanently "written in" to the design program rather than supplied as external data. These cable details are also given in Fig. 15.

5. Initial system. Complete details of the starting system - in this case the system as it was in 1955.

This was broken down into individual ring-main and spur units. For each unit details were given firstly of the number, sizes and locations of all transformers, and secondly details of the interconnecting cables and switchgear. Each cable was specified merely as a sequence of

map nodes, and an input subroutine was then used to compute the corresponding cable length and impedance, and to build up the unit admittance matrix and its inverse.

6. Load data. The year by year load demands were then specified. As described in section 4.2.1, the load substations were grouped by area according to the blocks of 11 kV load which they supplied. For each area the total load was specified, and also the location of any new substation within the area, together with details of its transformer ratings. Fig. 16 gives a year by year specification of the load demands. Also for each year an indication was given of whether a ring-main or spur policy was to be used for the development of new designs (section 5.2).

7. Alternative sequence requirement. The final figure in the input data gave the number of next-best design sequences required from the dynamic programming analysis.

7.2 Results - Design Aspects

7.2.1 System Design

The study was run for a design period of six years, with the data illustrated in Fig. 16. The starting system (corresponding to 1955) turned out to be good enough to meet the requirements of the next three years also. In 1959 three new substations required supplies (Fig. 16(b)) and the design algorithm produced four alternative new designs. All these designs were satisfactory for the following year also. In the

last interval of the design period, three more new substations required supplies. The design algorithm produced two new alternative designs from each of the four of the previous interval. This successive development of designs is shown in Fig. 17. In this figure each new system has been labelled with identifying suffices according to the notation of section 3.1 and Appendix A1.1; systems which have continued unchanged from the previous interval have been left unlabelled. (It must be emphasised that the lines connecting successive systems in Fig. 17 represent logical steps in the ad-hoc design process and not the only possible steps in a design sequence. In determining the optimal design sequence, the possibility of steps from any design in a given interval to any of the designs in the succeeding interval were of course considered - see for example Fig.19.)

As described in section 7.1 the choice of design policy (ring-main or spur) was controlled by the input data. Since in the corresponding real-life situation 1961 was the time of changeover from a ring-main to a spur policy, a corresponding change (i.e. for the last interval of the design period) was specified in the computer design study.

In order to reflect the continued running of the system beyond the last year of the design study (section 6.1.5), the last interval of the design period was specified as 10 years. (i.e. so that the length of the whole design period was comparable with the equipment lifetime - see also section 7.2.2.)

Figs. 18(a) to (m) show the starting system and 12 systems subsequently evolved by the design algorithm. (The

identifying suffices for each system refer to Fig. 17.) The basis for these diagrams was prepared by the computer on a line-printer. (A discussion on the use of graphical output will be given in section 7.3.2.) A key to the notation used appears on Fig. 18(a), which illustrates the starting system as it was in 1955. This system consisted of a single grid supply substation supplying three separate units, with transformers labelled A, B and C respectively. (A, B and C refer to units numbered 1, 68 and 119 within the computer. As each unit developed in time its unit number was constantly changed, but the same identifying letters are used in successive diagrams to provide a sense of continuity.) Units A and B were of ring-main type, but unit C (a single transformer only) was of spur type - one of the many anomalies which occur in real-life situations.

The four new systems designed to supply additional substations in 1959 are shown in Figs. 18(b)-(e). In the system of Fig. 18(b) the three new substations have been supplied by breaking open existing ring-mains and running out a double cable to include the new substations in the rings - one in unit A, two in unit B. For unit B this then meant that considerable overloads were incurred in the event of a cable outage, and a reinforcing radial feeder was run out from the grid supply point to one of the new substations, to reinforce the ring. In the system of Fig. 18(e) the new substations have been supplied by running radial feeders out from the grid supply point to the new substations, and then continuing

them to terminate on substations already in existing units. The systems of Figs. 18(c) and (d) are alternative combinations of these two developments of units A and B respectively. (All these designs are merely practical examples of the approaches discussed in section 5.)

Figs. 18(f)-(m) show the eight new systems developed in the final interval of the design period. This was the interval in which a change was made to a spur design policy. Figs. 18(f) and (g) show the two systems developed from system (4, 0) of the previous interval. In the system of Fig. 18(g) the three new substations, each with two transformers, have been supplied by running two spurs in parallel along the same route, each spur supplying one of the transformers at each of the substations. In Fig. 18(f) the new substations have been supplied by spurs run out, not from the grid supply point, but from the nearby existing substations in unit B. The result of this was to overload unit B, and a reinforcing radial feeder was added as shown. The other three pairs of systems are variations on these two designs, but with minor differences. For example, the system of Fig. 18(l) differs from that of Fig. 18(f) both in the method of reinforcement for unit B, and also in the choice of one transformer for an increase in size.

Having surveyed the range of systems evolved by the design algorithm the question naturally arises as to whether they resemble in any degree the systems actually developed. There was in fact only a slight resemblance. This was chiefly because the three substations added in 1959 were not completely

new substations, but were detached from another unit originally supplied from a different grid supply point. Since these substations were already connected together by cable, advantage was naturally taken of this cable in arranging a supply from the new grid point. The additional problem of "free" available cables could not be handled by the design algorithm (i.e. effectively a "major" design change). But on the other hand, the spurs in the systems designed for the 1961 interval which connected directly back to the grid supply point did reproduce exactly the real-life designs.

It must be concluded that the design algorithm as first developed was evidently of limited use for the production of useful designs, though it was encouraging to see that technically satisfactory designs could be produced, even if they were not economically satisfactory. Further work is obviously required on the development of the design algorithm. Possible lines of approach will be suggested in section 8.

Even with the small system examined in the first design study, problems arose concerning the interaction with the part of the system operating at 11 kV. There were several examples of the design algorithm demanding changes in the sizes of the 33/11 kV transformers, and in some cases (e.g. Fig. 18(b)) even changing the sizes of new transformers just specified by the input data. This would have a considerable effect on the design of the 11 kV system.

7.2.2 Costing and Sequence Analysis

Fig. 19 shows the corresponding system conversion and

operation costs (section 6.1.3). (Particular systems on the diagram may be identified by reference to Fig. 17.) The conversion costs are shown along the lines corresponding to changes between systems, and the annual operating costs next to the corresponding systems. The conversion costs usually exceeded the operating costs, except where a system continued in operation in successive years, and a zero conversion cost was incurred. In the final interval the operation costs had to be summed over a period of 10 years. The summed costs ("present worth" referred to the start of the interval) are shown beneath the corresponding annual costs.

The costs shown in Fig. 19 had to be corrected to their "present worth" referred to the start of the study before the dynamic programming analysis could be performed. The results are shown in Fig. 20, where the six cheapest design sequences and also the most expensive sequence are indicated. The figures inside the state circles show the cheapest costs involved in reaching each state. The lowest of these figures for the final interval determined the best (i.e. cheapest) sequence. The second best sequence shows a cost increase of 3.6%. It differs from the best sequence only in that one of the groups of new substations in the 1959 interval is met by a radial feeder arrangement rather than by tapping into the existing ring of a ring-main. Sequences 3 and 4 start along the 2nd best sequence and converge back onto the best sequence. Sequence 5 starts along the best sequence, deviating only in the last design interval. It is the first sequence to end in

a state in which the new substations in the final interval were supplied by running spurs to the substations of existing ring-main units rather than back to the grid supply point. Altogether the first 11 sequences gave costs within 10% of the best sequence.

As might be expected, the best sequence follows a natural path of logical ad-hoc development (see Fig. 17), as do the 2nd and 6th best sequences. It is to be expected that this would generally be the case. If a best sequence did deviate from the logical sequence this would be most likely at a time of major system reorganisation, when even the logical development itself would involve considerable changes.

Having obtained the best design sequences, the designer might well wish to perform further calculations, as discussed in section 3.3, to determine the stability of the solutions under the change of interest rates, load forecasts, etc. It would be useful if such computations could be performed as an integral part of the long-term design process. The designer could specify his requirements as part of the input data to the program.

An investigation was performed of the effect on the sequence analysis of specifying a final design interval of 15 years rather than 10 years. As was expected, the results were comparatively insensitive to this change. The sequence orders for the two cases (10 year interval first) were

1	2	3	4	5	6	7	8	9	10
1	2	3	4	8	6	7	5	11	10

It would be feasible to include such a check within the standard cost analysis routine. The check could be performed with a variation of ± 5 years on the specified final interval. This may be done with very little extra computation by re-costing the best 20 sequences. (The re-costing involves merely the addition or subtraction of a simple multiple of the corresponding operation costs for the final interval.) If these 20 sequences are then re-ordered, it may be safely assumed that the first 10 of these are in fact the best 10 sequences for the new conditions.

It would be unwise to draw any sweeping conclusions on the basis of such a small scale test, but the results obtained do seem to indicate the usefulness of the two-stage design approach.

7.3 Results - Computational Aspects

7.3.1 Basic Program Features

All the computation for the long-term design project was done on an I.C.T./Ferranti Atlas Computer. (The completed program was usually run on the "Atlas" of the National Institute for Research in the Nuclear Sciences at Chilton.) This machine had an operation time of about 2μ sec. for basic indexing instructions and floating point add/subtract, and about 5μ sec. for floating point multiplication. As far as a user was concerned the machine behaved as though composed entirely of single level fast access core store. To achieve this, exchanges between drum and core store were controlled automatically by a supervisor program. On the Atlas computer at

N.I.R.N.S. up to 100 thousand words of store were available.

The long-term design program was written in "Atlas Autocode", a language very similar to Algol. (An example of Atlas Autocode has already been given in Fig. 14 (b).) The entire program was expressed in approximately 4000 individual autocode instructions. About 24 million machine instructions were performed in the process of translating the autocode source program into machine code object program. The object program itself occupied 44 thousand words of computer store, made up of:-

- 13% main design control program (including input and output subroutines)

- 70% design routines

- 12% costing routines

- 5% dynamic programming (best sequences) routine.

The store occupied by the design portion of the program was apportioned between the three basic routines (section 5.2) as follows:-

- 9% "trach"

- 58% "menews"

- 23% "uloch"

- 10% other common routines

For the initial design study under discussion, the total storage required was 62 thousand words, made up of:-

- 44000 program

- 4000 compiler subroutines

- 14000 data and working space

A complete program run required approximately 5.6 minutes of Atlas computing time, of which 1.0 minutes was spent compiling the object program.

The above figures do not include the use of any graphical output facility. The routine which produced systems diagrams on the line-printer required a further 2500 words of program storage, with a corresponding increase in computing time.

7.3.2 Graphical output

Fig. 21 shows a typical extract from the program output obtained on a line-printer for one of the earliest computer runs. This extract shows some of the results for the 1959 interval. At the top of the extract are given details of a new unit, no. 464, replacing unit 1 of the old system. The first line of this unit description contains details of the number of transformers, type of unit (ring-main), grid supply point, and various items referring to the storage of the remaining information. The next line lists the map node numbers of the 5 transformers, followed on the next line by a list of transformer sizes (size 1 \equiv 5 MVA rating). Then follow 14 lines, grouped in pairs, each pair describing one cable of the unit. The first line of each pair gives details as to which two points the cable connects, the switch-gear layout at each end of the cable, and storage details of the corresponding actual cable route, which is given node by node on the following line. At the bottom of the extract each of the four new systems designed for this interval is checked for transformer and cable overloads. The caption "CHECK NEW SYSTEM" is

followed by a list of the numbers identifying the units which make up the system. Details of changes of transformer sizes and laying of reinforcing cables then follow, together with an indication of the corresponding changes in unit numbers. When all four systems have been checked, the program continues to the next design interval, and the final line of the extract shows that it was about to commence the design process on the first of the four systems designed in the previous interval.

Even for the small design study under examination it soon became apparent that the presentation of results in the form of Fig. 21 was not good enough for practical use. The large quantity of numerical information still had to be analysed and re-presented in some more digestable form. In particular there was evidently a great need for some form of graphical output. The labour of taking all the cable routes for a system, presented in the form of sequences of map node numbers, and plotting out the actual routes on a map was especially laborious and time consuming.

The ideal solution was to use a proper C.R.T. type graphical display. There was a Stromberg-Carlson SC 4020 display unit available (at the Atomic Weapons Research Establishment at Aldermaston). This device plotted diagrams on a 1024 x 1024 raster (i.e. matrix of individual points), and was provided with vector and character plotting facilities. However, the SC 4020 was designed to work from an I.B.M. magnetic tape unit, and no software support for the production of such tapes was available on the Atlas. A project was set

in hand to write programs for the Atlas which would generate the required command instructions for the SC 4020, and pack them on I.B.M. magnetic tape in the required format. Whilst this was viewed purely as a long-term project, program development at the time of writing this thesis is practically complete, and the SC 4020 should be available for use with later design studies.

In the short-term it was decided to try to use the line-printer to obtain diagrams of the distribution systems. This project had the virtue of comparative simplicity. Also this form of output could be obtained from any Atlas, whereas facilities for the use of I.B.M. magnetic tape were limited to particular machines. A standard line-printer had a maximum line length of 120 characters (at 10 per inch), thus the diagrams had effectively to be plotted on a 120 x 120 raster. This was considerably less accurate than the SC 4020 display. A complication arose from the fact that the vertical line spacing was 6 per inch, requiring a different scaling along the vertical axis, and even further reducing the accuracy of the diagram. The problem of accuracy was somewhat relieved by using the full stop and apostrophe characters superimposed in one character position (see Fig. 18(a) where a portion of cable route has been left in its original form, direct from the printer).

The diagrams of Fig. 18 were produced on a scale of 1 inch to 800 yards. This was the smallest scale on which the cable routes along individual roads could be represented with any

accuracy. Whilst the systems under discussion could easily be represented on a 12 inch paper width, this would not be the case for the whole of the Edinburgh system, which will require the use of the SC 4020 display if satisfactory diagrams are to be obtained.

Details of switchgear installations are not given on the diagrams, but must be obtained from the conventional output of which Fig. 21 is an example. No indication is given of how many cables lie along a particular cable route, but this may usually be deduced by inspection.

The map drawing subroutine was inserted into the design program so that a diagram was produced of the starting system, and of every new system after all the design changes had been made at the end of each design interval.

7.3.3 Store and Instruction Economy

The initial "de-bugging" runs of the program required a considerable amount of computing time - in excess of 4 minutes to complete even a small part of the design phase of the study. To reduce the computing time, some of the most frequently used routines (matrix inversion, matrix correction, load flow) were re-written in machine code. The hand-coded routines showed savings of as much as 3.5:1 on the corresponding number of instructions in the translated versions of the autocode routines. This resulted in a marked reduction in computing time (although no precise measurements were made).

The program made considerable demands on computer storage also (details are given above). Although the requirements

for the first design study were well within the capacity of the N.I.R.N.S. Atlas, they were nevertheless considerable when related to the very small size of the problem. Evidently any attempt at a problem of reasonable size would strain the available storage to its limits. Of the 14000 words of data and working space, 10000 words were used for the storage of the information describing the individual ring-main and spur units. Investigation showed that a considerable improvement could be made in the manner of unit storage.

There were two main sources of inefficiency. Firstly, when a new unit was created, all the information describing that unit was entered in the unit information list. This was obviously wasteful if the new unit had been formed merely by changing one transformer in an older unit. In this case all that was really needed was a new list of transformer sizes.

Secondly, every new unit was stored in the unit information list as it was formed. But consider the situation where the "menews" design routine had put forward a new design which had still to be checked for transformer and cable overloads. Suppose that one particular unit had a number of overloads. Each time a transformer size was changed, or a cable was added, a new unit was formed and the corresponding information stored. Thus a whole series of new units could be formed in the course of evolving one satisfactory design. Although only the last of the series was referred to in the system list, all the redundant intermediate units were still held in the unit information list. (The first unit put forward by

"menews" was not redundant, however, since this was referred to by other new systems formed at the same time.) The storage system as described (section 5.2.1 and Fig. A3 of the Appendix) is obviously capable of considerable improvement - some suggestions will be given in section 8.

8. Suggested Future Developments

"Its highest solution must be evolved from the eye and brain and soul of a single man, which from hour to hour are making subconsciously all the unweighable adjustments, no doubt with many errors, but with an ultimate practical accuracy."

- The art of war at the beginning of the 18th century, described by Sir W. S. Churchill.

It is apparent from the previous section that much remains to be done in the practical study of the use of the two-stage design algorithm. Some of the steps which would be involved in carrying out a full study are discussed below:-

8.1 Short-Term Improvements

Certain features of the existing computer program require immediate development. Perhaps the most important of these is an improvement in system storage arrangements. The difficulty of redundancy within units could be overcome by only slight modifications to the existing scheme of storage. The original storage layout for a single ring-main unit is illustrated in Fig. A3(b) of the Appendix. The term "link" is used to denote a word of store holding the address of some information of interest. "Links" were used in the original storage scheme to provide quick access to various items of information. But they could point only to store locations associated with their own unit. In the modified system of storage suggested in Fig. A3(c) the use of links has been considerably extended. A link would now also be permitted

to point to locations originally associated with an earlier unit. Thus if a new unit was created by a single modification of an earlier unit, most of the links in the new unit could still point to the unchanged information of the old unit. For example, a ring-main unit with 4 transformers and 5 cables would require 67 words of store on the old storage scheme, 71 on the new scheme, if both were completely new units. But for a unit developed from an earlier unit by the change in size of a single transformer, the new scheme would then require only 17 words.

The problem of completely redundant units is not so easily dealt with, and would be better left for later consideration.

There would also be much to be gained from a general increase in the computational efficiency of the program. In the short-term this could be realised by continuing the policy mentioned in the previous section of replacing the autocode versions of the most frequently used routines by more efficient machine-coded versions. The obvious candidates for improvement in this way are the store mapping functions (section 5.2.1), and also the "shortest path" routine which determines the best routes for cables.

With these improvements it should be possible to test the design algorithm with a larger test case involving the whole of the Edinburgh system and covering a period of the order of 7 years.

8.2 Medium-Term Improvements

8.2.1 Computational Aspects

The short-term improvements are concerned only with the more practical aspects of computation, to allow further testing of the design algorithm to proceed immediately. This effort should also be continued in the medium-term studies.

However good may be the storage arrangement finally chosen, it now appears that even with the large amount of effectively "one-level" store available to users of the Atlas computer, some form of auxiliary storage will soon be required. With the Atlas this will probably take the form of magnetic tape. The logical extension of this step would then be to segment the program completely, and run the second stage costing algorithm separately, using the system information on magnetic tape as data. This would provide operational advantages. Also, the new systems would be available if later tests were required on the stability of the best sequences with respect to various input parameters (sections 3.3 and 7.2.2).

At the same time careful consideration should be given to the use of a complete list processing language of the "LISP" type (see for example [31]), to replace the store mapping functions and list arrangements of the original program. Languages of the LISP type are designed specifically for the efficient use of computer storage, and may include special facilities to deal with the removal of redundant information. The best solution might be a general purpose language of this type especially adapted to take advantage of

the particular features of the long-term design problem.

A further increase in computational efficiency might be obtained by complete reorganisation of the use of the "shortest path" routine. One extreme possibility would be to compute once and for all every possible cable route, (i.e. from every map node to every other map node), and to store these on magnetic tape. A more likely compromise would be to compute every route as it was required, and then to store the results (in fast access store) in case the same problem should arise again. It is extremely likely that the same problem would recur frequently, as may be seen from the diagrams of Fig. 18, particularly within the same design interval. At the end of each interval the old library of routes could be discarded and a new one built up for the new problems of the next interval.

8.2.2 Design Aspects of the Edinburgh System

It is obvious from the results obtained from the first program runs that considerable improvements must be made to the actual system design routines, if results are to be produced which will be of real interest to a design engineer. The original expectation that most of the programming effort would be required on the ad-hoc design process is amply borne out by the figures given in section 7.3.1. Thirty thousand words of store were required for the design algorithm alone. Some of the difficulties were due to the peculiar nature of the systems being studied, but others would be common to most programs based on an ad-hoc design approach. Perhaps the chief difficulty was that of operating on geographical problems.

Whilst eye and brain seem very adept at processing much visual information simultaneously, the reverse is true of a serially operated digital computer.

Programming was further complicated by the use of two separate design policies. In practice the ring-main policy seemed to involve more "common-sense" and experience, whereas the spur policy needed less "common-sense" but presented more alternatives from which to choose. Thus the latter policy was more suited to programming on a digital computer than the former. Furthermore, since the spur policy is now the only one in use, it would be more profitable in every respect to restrict program development to this policy only, and to continue with investigations of future system development rather than with retrospective studies.

One possibility is to continue the development of the existing spur design routines to a more sophisticated level. As well as general all-round development, four specific points require attention:-

1. Production of Alternatives. In the original program, it had been decided (for the purpose of simplification) that only the design routine dealing with the laying of supplies to new load substations should put forward a number of alternative designs (section 5.2.4). The facility of producing a number of alternatives should be extended to the other two basic design routines dealing with the correction of transformer and cable overloads.
2. Interdependence of Design Sections. It had been

assumed that the three basic design processes, embodied in the corresponding design routines (section 5.2), were independent. In fact these processes are not completely independent. In particular, the way in which supplies are laid to new substations depends to some extent on the loading of the cables in existing units. For example, it may not be profitable to divert a cable which is loaded to capacity to supply a new substation, even if the diversion would be a very short one. This aspect should be included in the design algorithm.

3. Sub-optimisation. Another feature of the original design program was that the designs produced in any one interval were decided purely by the load demands of that interval. On the large scale, the job of taking into account future load demands in selecting designs for earlier intervals is done by the second stage costing analysis. On the small scale, however, limited consideration of future loads could profitably be written into the design routines of the program. For example, instead of laying a cable directly along the shortest path between two substations, it might be worthwhile to divert it to pass the site of a new substation to be introduced in the following year. This is essentially a process of sub-optimisation, which should operate only over a very short time scale (two or three intervals within the design period).

4. Improved Spur Design A description is given in Appendices A2.2 and A2.3 of how the routine for the

design of new spurs could be extended to include more realistic situations. With the added emphasis on spur systems, these suggestions should certainly be implemented.

The four points so far mentioned are to some extent, of lesser importance, and may be implemented whenever convenient. One design problem, however, is of far more pressing importance. It has been emphasised that the design routines as written were unable to deal with design situations in which a major system reorganisation was required. There are three reasons why such an extension to the program should be given a high priority.

Firstly, such design situations must inevitably be dealt with if design studies are to be made over realistic periods of time (e.g. of the order of 15 years). Secondly, the possibility of using such times of major reorganisation to limit the growth in the number of new designs (section 5.1.3) has yet to be investigated. Thirdly, it is at such stages in the design (frequently associated with the addition of new grid supply points) that the maximum interaction takes place between parts of the system of different voltage levels. This important topic will be discussed below.

Careful consideration should be given to the method by which it is intended to tackle such design problems. An obvious approach would be to develop ad-hoc design routines along similar lines to those already used for the less complex design situations. This would require a major effort of design study and programming, though the problem would be considerably

eased if attention was confined to spur designs only. It might be possible to save programming effort (at the expense of computing time and storage) by writing a more general type of design program which produced a very large range of possible designs. As mentioned in section 5.1.2, the dynamic programming analysis would eliminate the uneconomic designs, provided there were a few good ones amongst them. This design development would be best undertaken by a design engineer with some programming knowledge, rather than a specialist programmer.

As a short-term 'stop-gap' it might be possible to avoid writing new design routines to deal with major system changes by running the existing design program in sections. Each section would start with the design interval following a major reorganisation, and continue with modified designs until a new reorganisation was required. The results would then be returned to the designer who would put forward his own plans for the reorganisation, and supply his new systems as data for the next run of the design program. It should be emphasised that this possibility should only be considered as a temporary expedient. It should only be undertaken if computer facilities were available which offered the prospects of a "turn-round" time for programs of the order of one day or less. A more sophisticated variation of this approach is discussed in section 8.3

8.2.3 General Aspects of Two-stage Design

The programming improvements suggested above are aimed specifically at the production of useful results relating to

the design of the Edinburgh distribution system. Also of importance is the need to continue development and to gain experience in more general aspects of the application of the two-stage design method. Four facets are of particular importance:-

1. Multi-level systems. The need to investigate the inter-actions between the particular system under study and external systems linked with it has already been mentioned above. A description was given in section 4.2 of how the distribution system at 33 kV was isolated from the systems at higher and lower voltages by the device of using the description of these systems as fixed data for the design of the 33 kV system. The next step is evidently to advance to the design of a two-level system. For the actual problem in question this may probably be done most easily by considering also the design of the 275 kV system. This would involve a relatively small quantity of additional information, as opposed to the 11 kV system, which is very much more complex than the 33 kV system. Furthermore, it would appear that any study of the future Edinburgh system would in any case be closely linked with the design of the grid supply network.

The design of the grid network might be included most easily by merely listing a number of sites for the supply substations, together with the total additional costs incurred in making power available at each one, and the corresponding maximum load they could supply.

Then in the course of the design procedure, at any interval in which the existing supply substations were shown to be overloaded, an additional supply substation could be selected from this list, and made available to the 33 kV system design routine. In the first instance it might be convenient if only one possibility was put forward at this stage. Subsequently this restriction could be relaxed so that at each step a number of alternatives were put forward.

The storage arrangements would have to be extended to cover the description of the distribution systems at both voltage levels. The costing analysis would also require extension so that the best design sequences for the combined systems could be determined.

2. Use of Several Development Policies. In the design studies so far performed, either the ring-main or the spur development policy has been specified for each design interval by control from the input data. The next development is to allow two (or more) policies simultaneously to put forward alternative designs in each interval. If the suggestions given above are followed, then there will be a change of emphasis from the previous contrast of spur and ring-main policies to a range of policies all based on spur units.

A first step would be to allow two design policies to put forward alternative designs, but confine the use of each policy to systems previously developed by that same policy - Fig. 22 (a). (The lines connecting design

states in Fig. 22 again represent steps in the ad-hoc development, and not the only possible steps in a design sequence. e.g. compare Figs. 17 and 19.) Thus the design algorithm would put forward for each interval designs of two distinct classes formed by the use of the two alternative policies.

The next step would be to allow the development of mixed systems. i.e. systems developed by the use of different policies at different intervals in their histories. The design algorithm would thus put forward not only two distinct types of systems, but also systems of intermediate types. Common sense suggests that in practice it would only be necessary to consider systems in which there had been only one change of policy during the succession of design modifications. (For example, a system derived by using two different policies in alternate years would be most unlikely to be economic.) This would give rise to the possible ad-hoc growth of systems illustrated in Fig. 22(b).

3. Avoidance of Redundant Computation. It is evident that when all the features so far described have been implemented, the program will require considerable quantities of computing time. It will be of great importance that the processes of computation should be as efficient as possible. When a system is being modified by an ad-hoc procedure, cases will frequently arise in which a considerable proportion of the particular local

problems encountered will also occur in identical form in similar systems arising from common roots in the ad-hoc design sequence. For example, in the last interval of the design study discussed in section 7, the pairs of spurs connecting the three new substations back to the grid supply point were designed 4 times. Since the pair were identical, the same problem was effectively solved 8 times.

Not only might computing time be saved, but storage also, if the recurrence of similar situations could be recognised, and the corresponding design information stored once only. On the other hand much temporary storage and extra computation would be involved in the search for common situations, and this would have to be weighed against any possible savings.

The way in which a scheme to take advantage of this feature might be implemented is by no means obvious. The scheme suggested in section 8.2.1 for the storage of cable routes in case they were required again is effectively tackling the same type of problem, and a similar approach might be tried in this case.

4. Generalised Long-term Design Program. The principle of two-stage design should be of interest over a wider context than the design of the Edinburgh distribution system. It would therefore be of advantage to make available a general purpose version of the long-term design program for use in other problems. This should

be done on two levels. A completely generalised version of the program would consist of the dynamic programming routine for sequence analysis, together with the bare outlines of the main design and costing control routines. It would then be left to the user of this program to specify his own storage arrangements and write routines to deal with the design and costing processes.

On an intermediate level, a general program for use in electrical network design would also be of use, incorporating much of the design program as it was originally written. The concept of a "unit" would be common to most of these problems, so the same storage system could be retained, but with scope for the user to redefine the store mapping names to suit his own problem. A large part of the present costing routines could also be incorporated. Many other routines - "shortest path" for cable routes, unit input routine, graphical output routines, matrix inversion and modification and load flow routines, also the "laynewspur" routine for the design of new spurs - would form a useful library of routines for use in the design stage of the program.

8.3 Long-Term Improvements

The difficulties of writing ad-hoc design routines have already been commented upon. Experience of tackling this problem points to the conclusion that this might be a case in which a man and a machine (i.e. computer) together might be

far more effective than either alone. In the light of the comments originally made at the end of section 2.3.3, it is interesting to see how this conclusion has been forced upon the author.

It is clear that the success of such a combination would be determined largely by the power and flexibility of the means of communication between the two. It is only the more recent advances in man-machine communications that have made the possibility of an efficient partnership between the two at all feasible. The first essential in such a situation is that the designer should have a powerful computing system immediately available to him. It must appear as if the computing system is available "on-line" for his personal use. It is to be hoped that after the pioneering work on Project MAC at the Massachusetts Institute of Technology (see for example [32]) such facilities will soon be generally available.

The second important requirement is that there should be some means of exchanging information between the man and the machine in a form which can be readily assimilated by both. In distribution system design much of the system information is topological and geographical. It is therefore logical to demand some device capable of visual input/output via which the man and machine may communicate. Fortunately such devices have recently been made commercially available - though much research is still needed into the most efficient ways of using them.

Such a computing system as has just been envisaged would

enable a fresh approach to be made to the problem of ad-hoc design, and hence to the use of the two-stage design algorithm. In this situation there would be no need to evolve long and complex design programs, and the two-stage design algorithm could be implemented with relatively little time-lag. In any particular case, the machine would display the actual system situation to the designer, who would draw in pictorially a number of possible alternative developments. There would be no need to ensure that all his suggestions were technically sound. It would be left to the computer to perform the necessary checks (e.g. load flows, short circuit analyses, etc.) on the suggested systems, possibly accepting some, rejecting others, and referring marginal cases back to the designer. It might still be worthwhile to develop sufficient design programs for the computer to tackle certain standard recurrent situations (e.g. reinforcing a ring-main with a radial feeder from the supply substation) without referring to the designer.

Another advantage of the "on-line" use of a computer would be that the designer would develop a far deeper understanding of the problem he was tackling. He would be made more aware of the significance of various factors in the design process by being presented not merely with the end results of a long process of computation, but by being able to monitor the intermediate steps also. The ease with which new ideas may be investigated should encourage the designer to explore the more unorthodox possibilities which he would previously

have neglected as being less likely to repay the labour of examination.

Lest these prospects should seem to be only dreams of the distant future, it should be pointed out that a comprehensive project on the use of a visual input/output device is scheduled to start shortly within the author's own department. The prospects of the University of Edinburgh obtaining powerful "on-line" computing facilities are also extremely promising. Provided both these projects progress satisfactorily, it seems quite feasible that at least a limited form of man-machine on-line implementation of the two-stage design algorithm could be brought to fruition within the next 2-3 years.

9. Conclusions

It is felt that the two-stage design approach may, for certain long-term design problems, provide a useful compromise between methods of exhaustive or intuitive search, and methods in which the problem is viewed purely in terms of the mathematics of optimisation.

The first design study has given encouraging results, but a much more intensive investigation is required. Continued development is particularly necessary in the design phase of the program. It is recommended that this should be directed towards a study of future system growth, rather than retrospective studies.

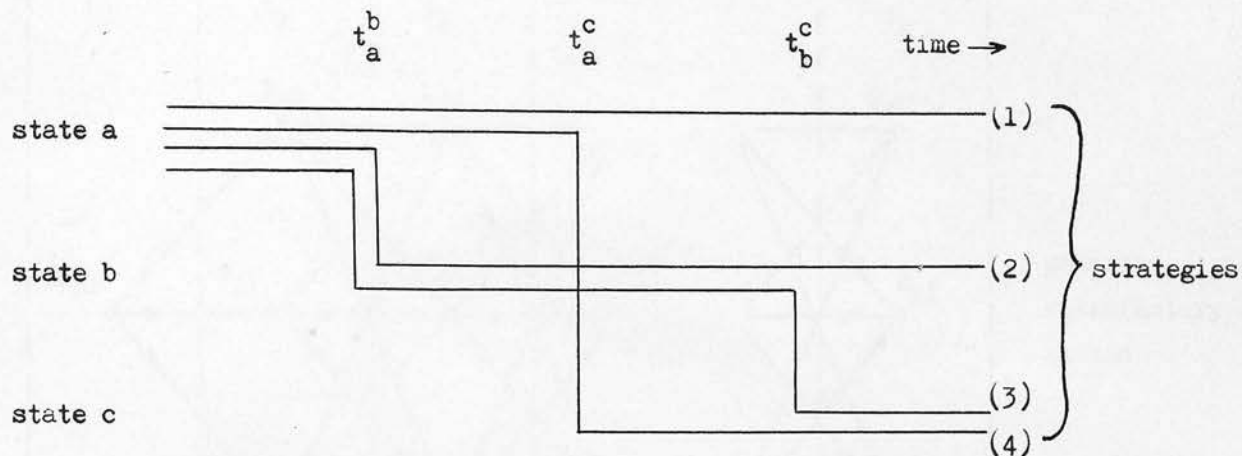
It is possible that the full potential of this approach to long-term design can only be fully realised in conjunction with a powerful "on-line" computing system incorporating good visual input/output facilities.

10. Acknowledgements

I would particularly like to thank my supervisor, Dr. J. V. Oldfield, for his continued advice and encouragement.

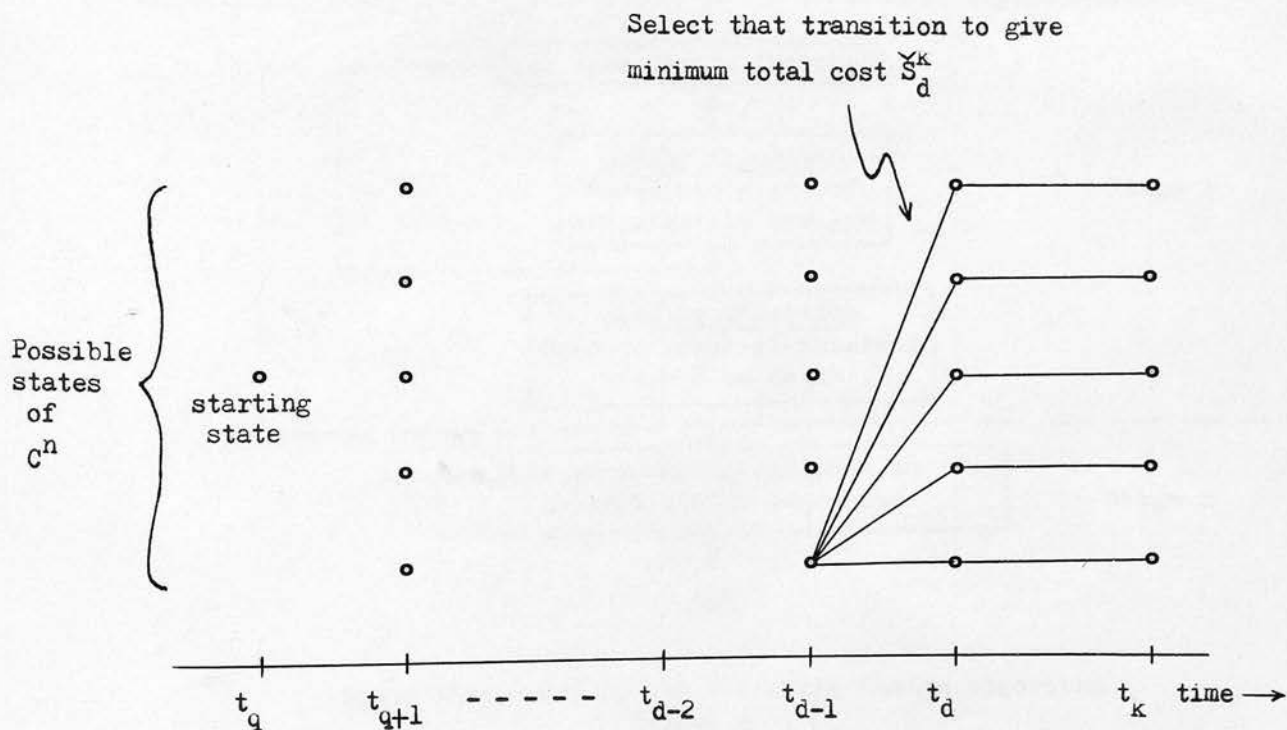
The computer time for this project was sponsored by the South of Scotland Electricity Board. Thanks are due especially to Mr. W. L. Kidd, Chief Research and Development Engineer, and also to Mr. W. Wood for much practical advice on system planning.

I am grateful to the S.R.C. (formerly D.S.I.R.) for the studentship which enabled me to participate in this project.



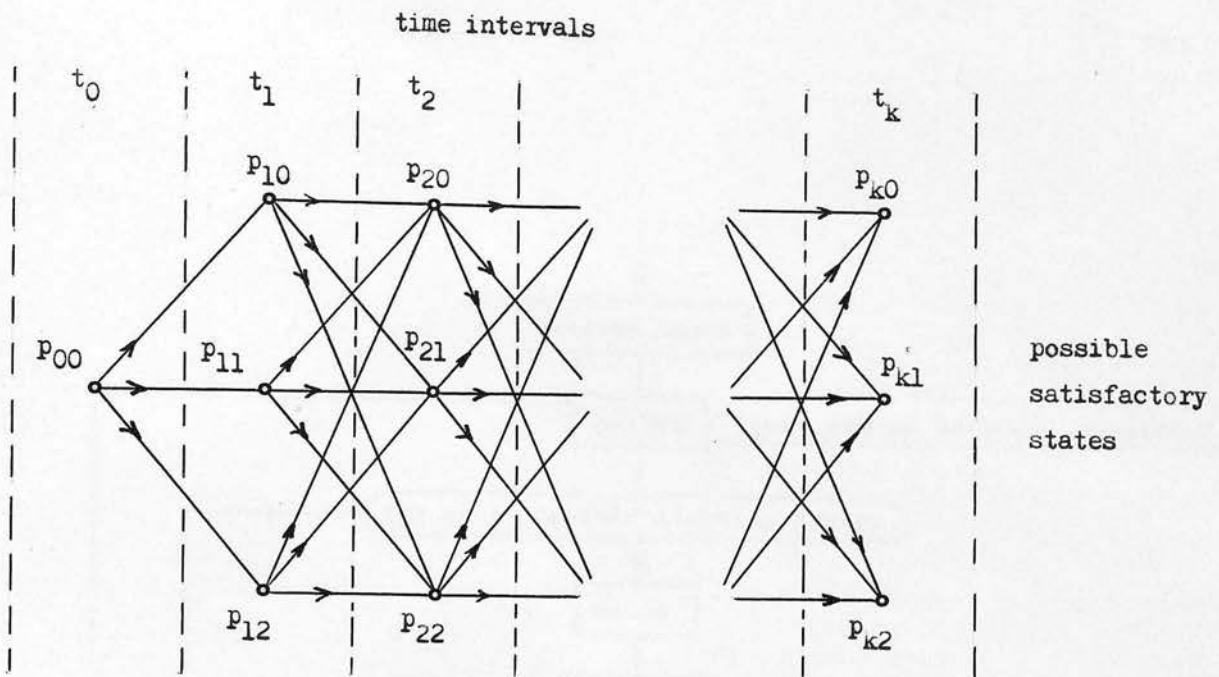
Possible strategies in the development of a single dipole

Figure 1



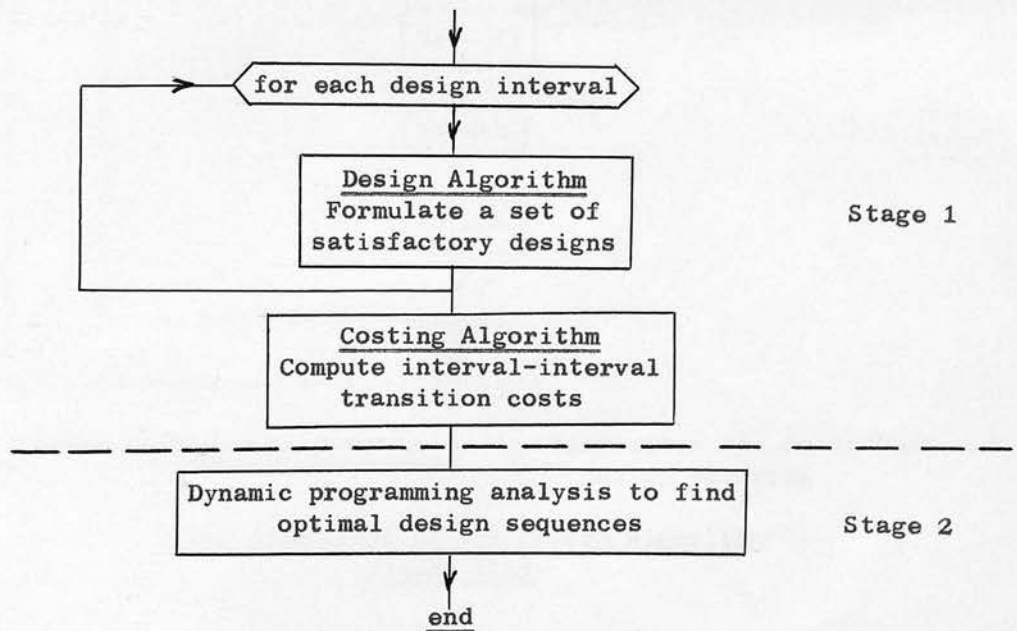
First stage in the optimisation of a parameter sequence
by dynamic programming

Figure 2



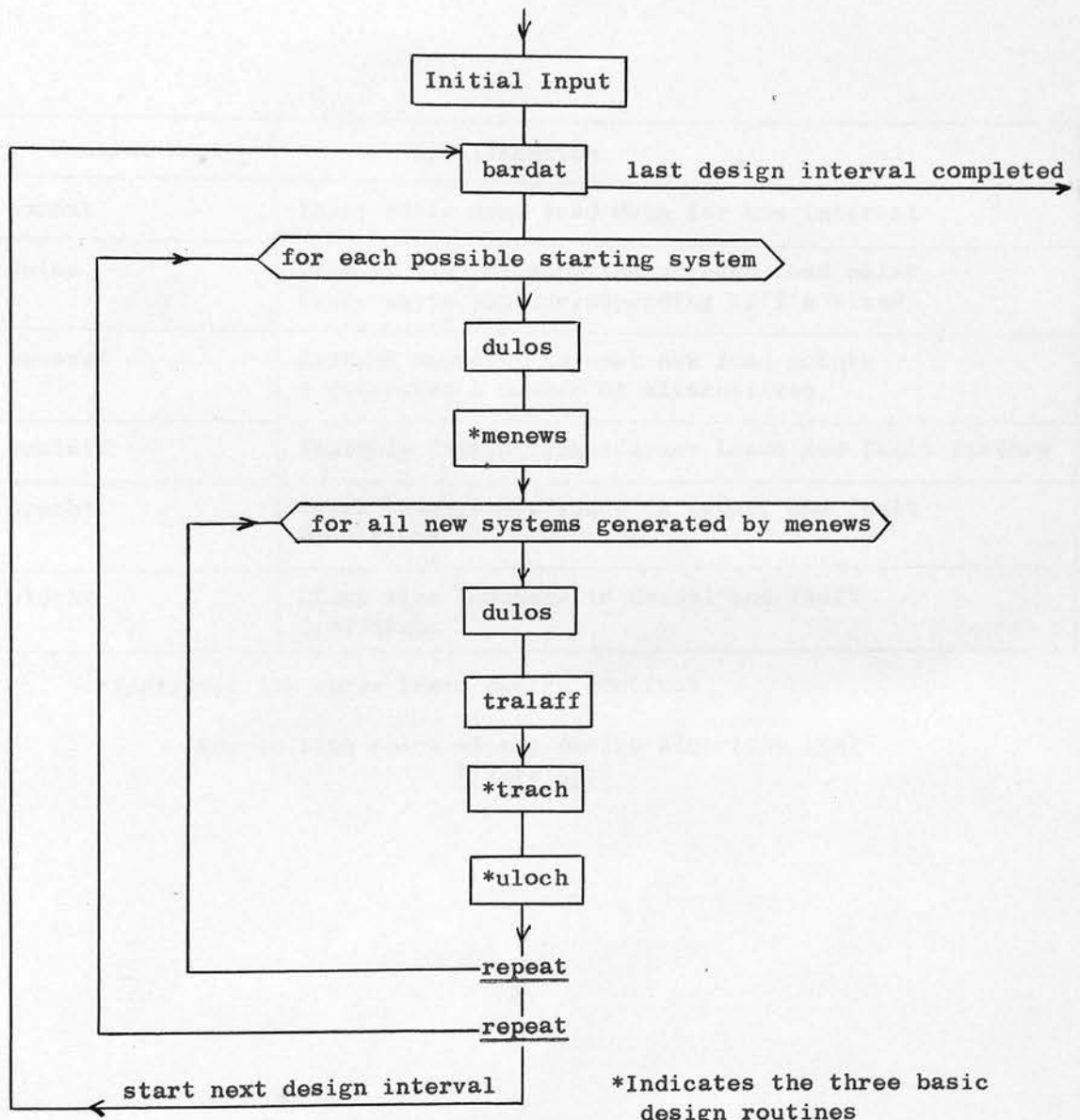
Possible sequences of design states

Figure 3



Basic flow-chart of the two-stage design algorithm

Figure 4



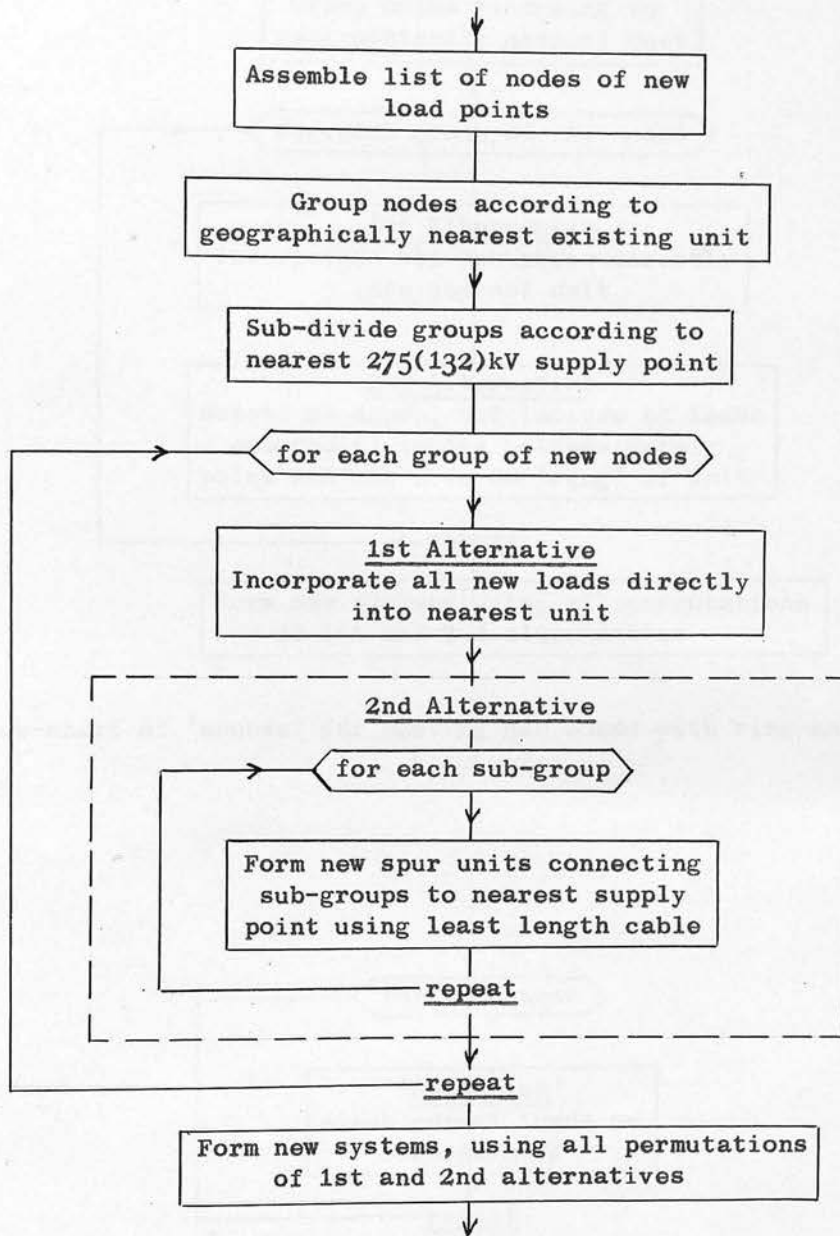
*Indicates the three basic design routines

Basic flow-chart of the design algorithm
Figure 5(a)

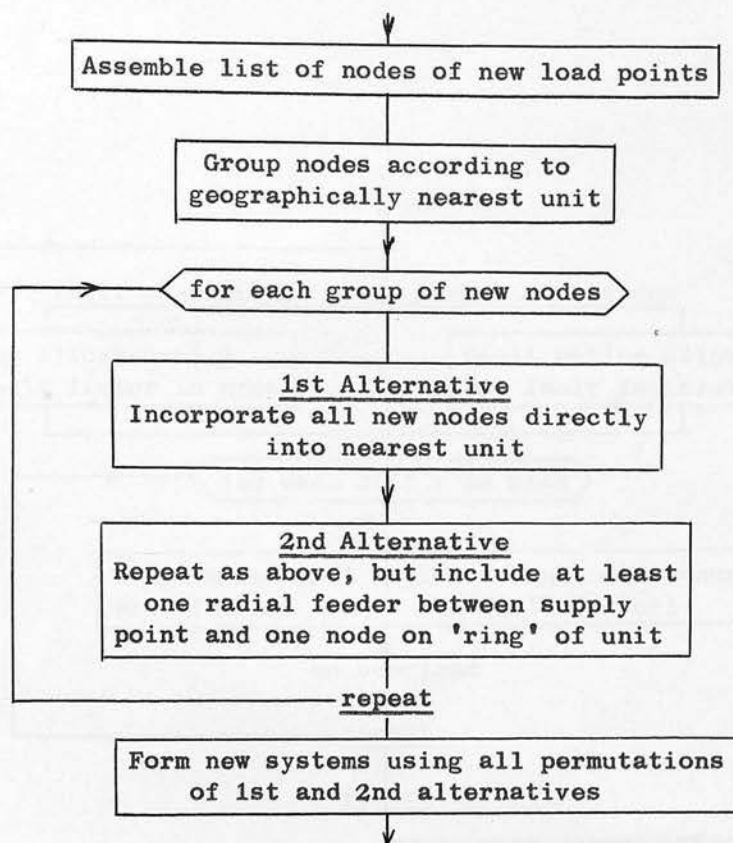
Routine Name	Specification
bardat	Input basic area load data for new interval
dulos	Draw up list of supplies at each load point (i.e. units and corresponding tr'f'r sizes
menews*	Arrange supplies to meet new load points - generates a number of alternatives
tralaff	Assemble list of transformer loads and fault factors
trach*	Check transformer loads in normal and fault conditions
uloch*	Check line loadings in normal and fault conditions

*Indicates the three basic design routines

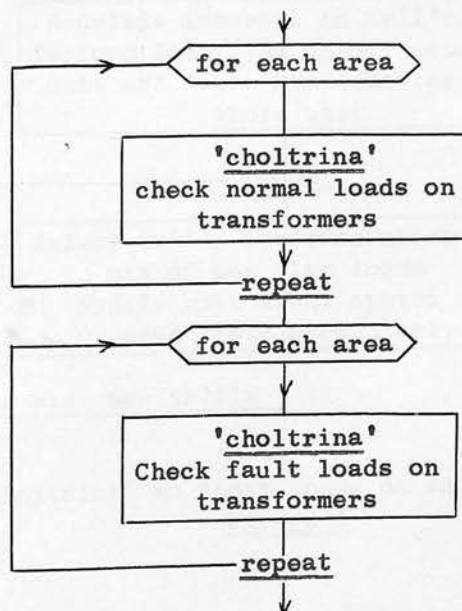
Key to flow chart of the design algorithm (5a)
Figure 5(b)



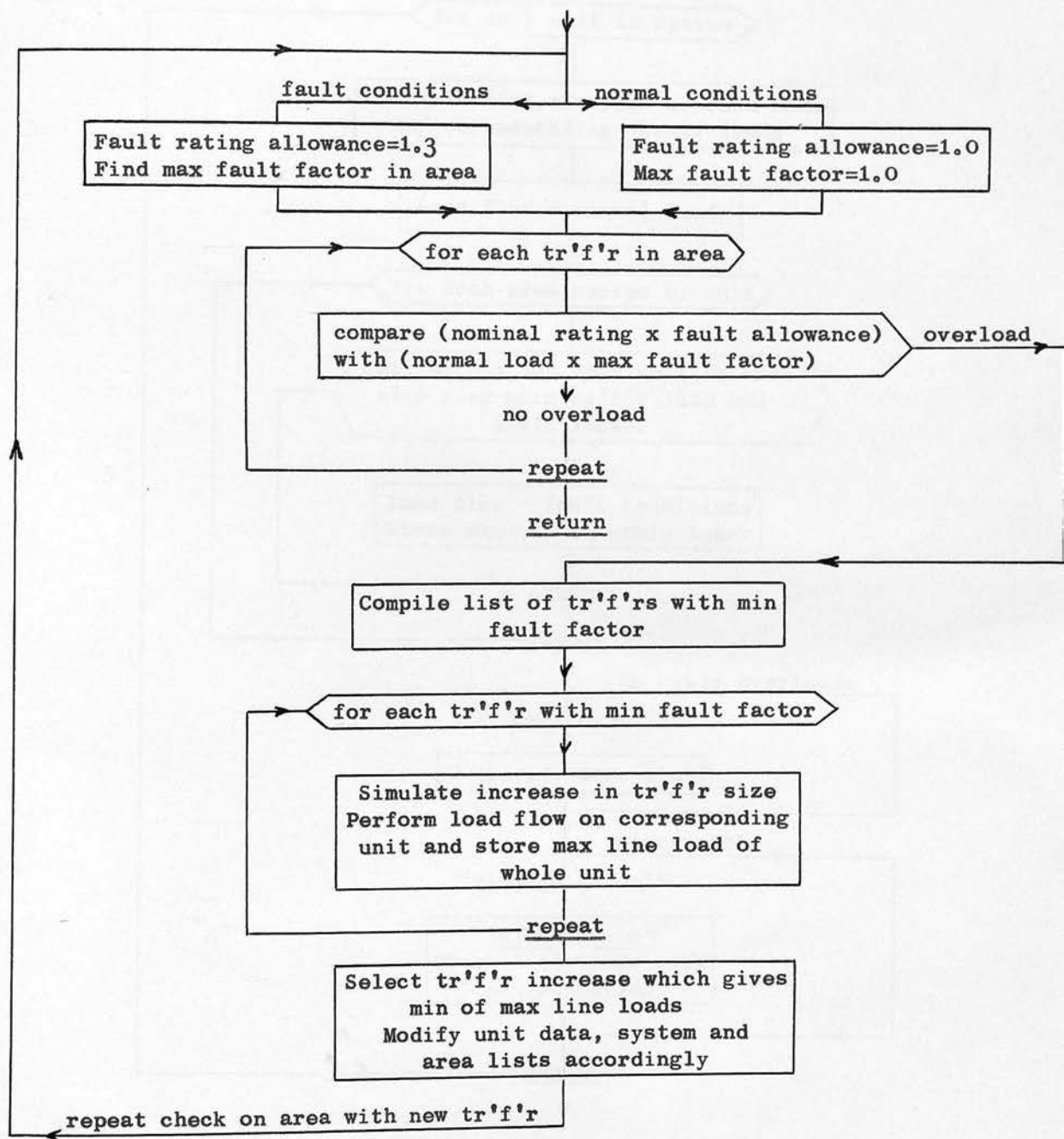
Flow-chart of 'menews' for meeting new loads with spur designs
Figure 6



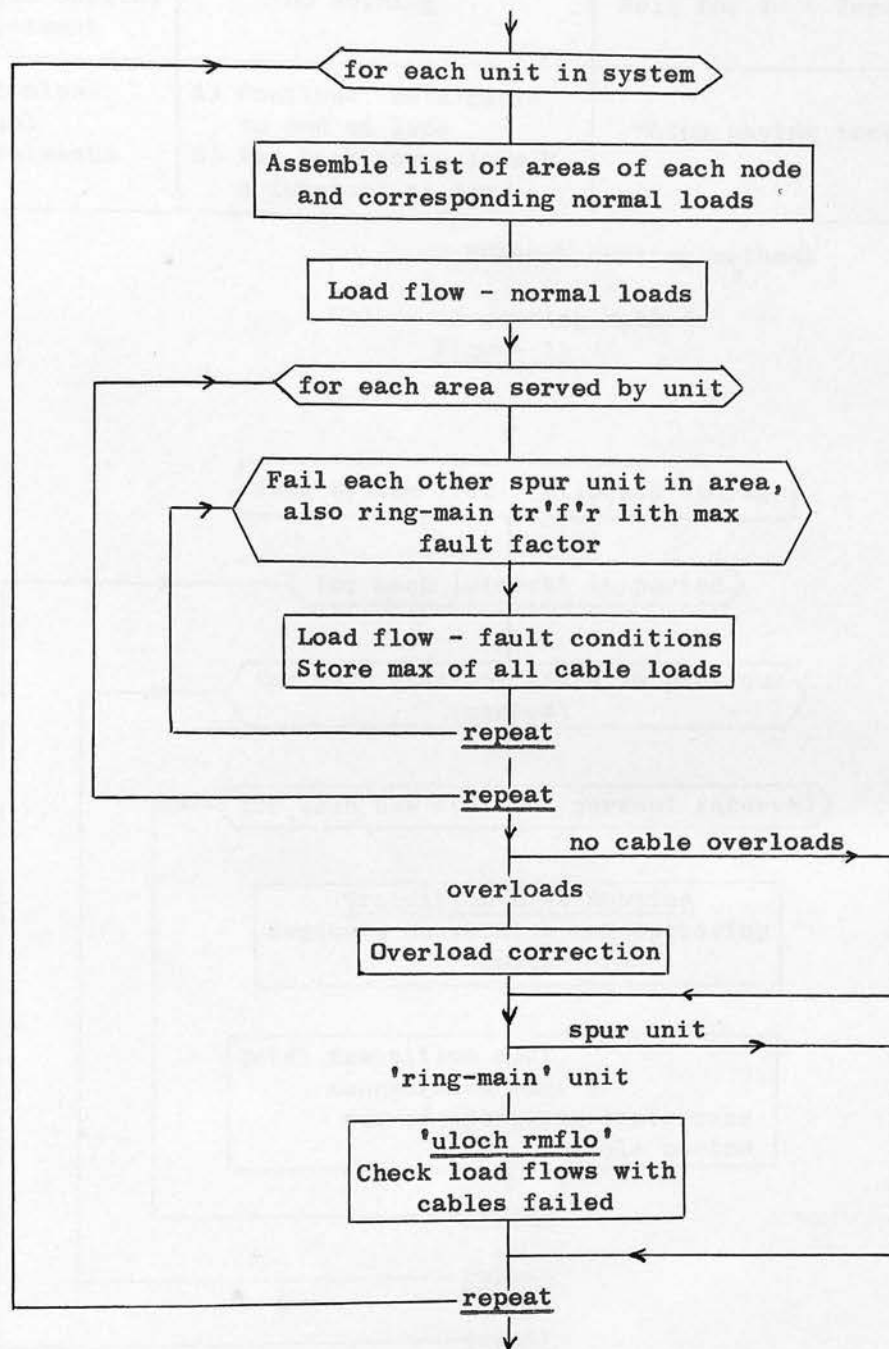
Flow-chart of 'menews' for meeting new loads with ring-main designs
Figure 7



Flow-chart of 'trach' to check transformer loads
Figure 8



Flow-chart of 'choltrina' to check loads on all transformers in an area
Figure 9

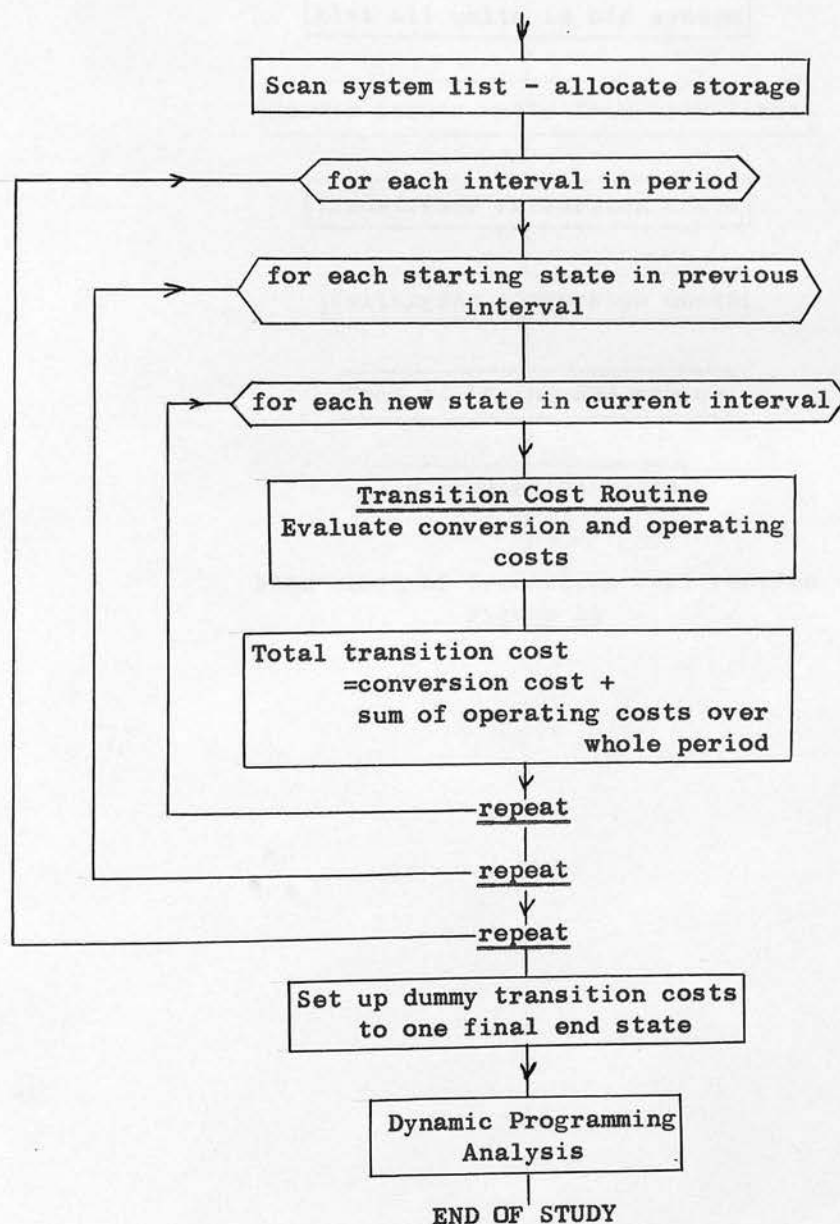


Flow-chart of 'uloch' to check loads in unit cables
Figure 10

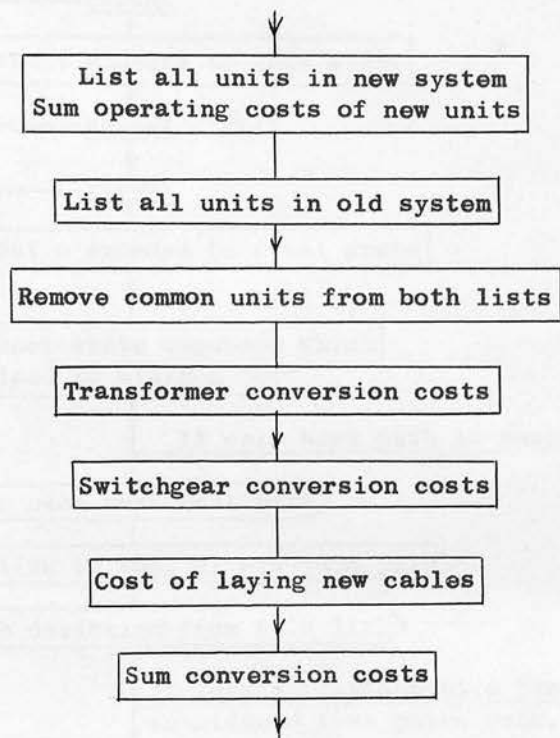
Method of Costing	Equipment discarded at age t	
	Type (a) Abandoned	Type (b) Re-utilisable elsewhere
1. Single Capital Investment	*Do nothing	Sell for V - function of age
2. Equivalent Annual Instalments	1) Continue instalments to end of life 2) Pay back now - loss V a function of age	*Stop paying instalments

*Chosen costing methods

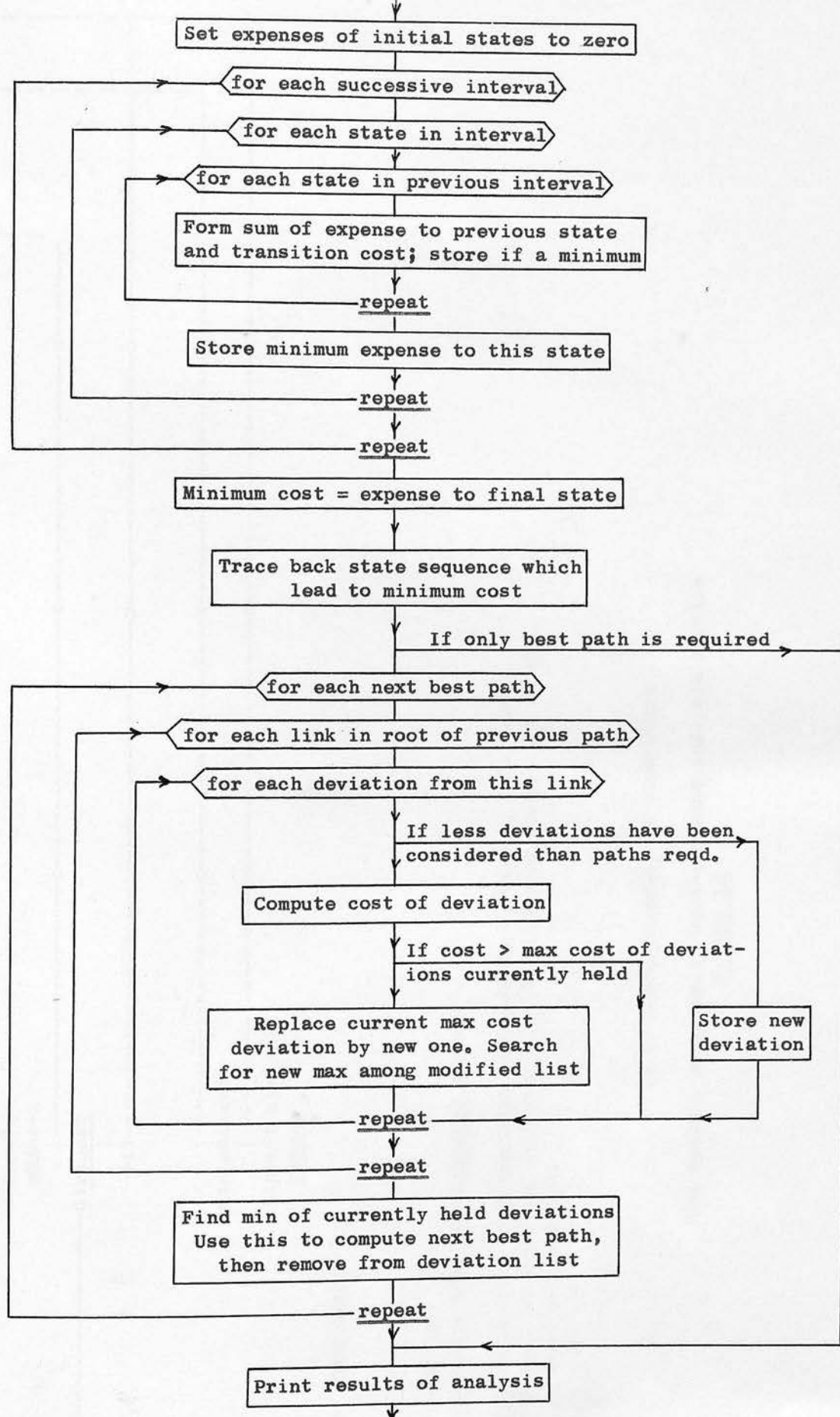
Choice of costing methods
Figure 11



Flow-chart of cost analysis routine
Figure 12



Flow-chart of transition cost routine
Figure 13



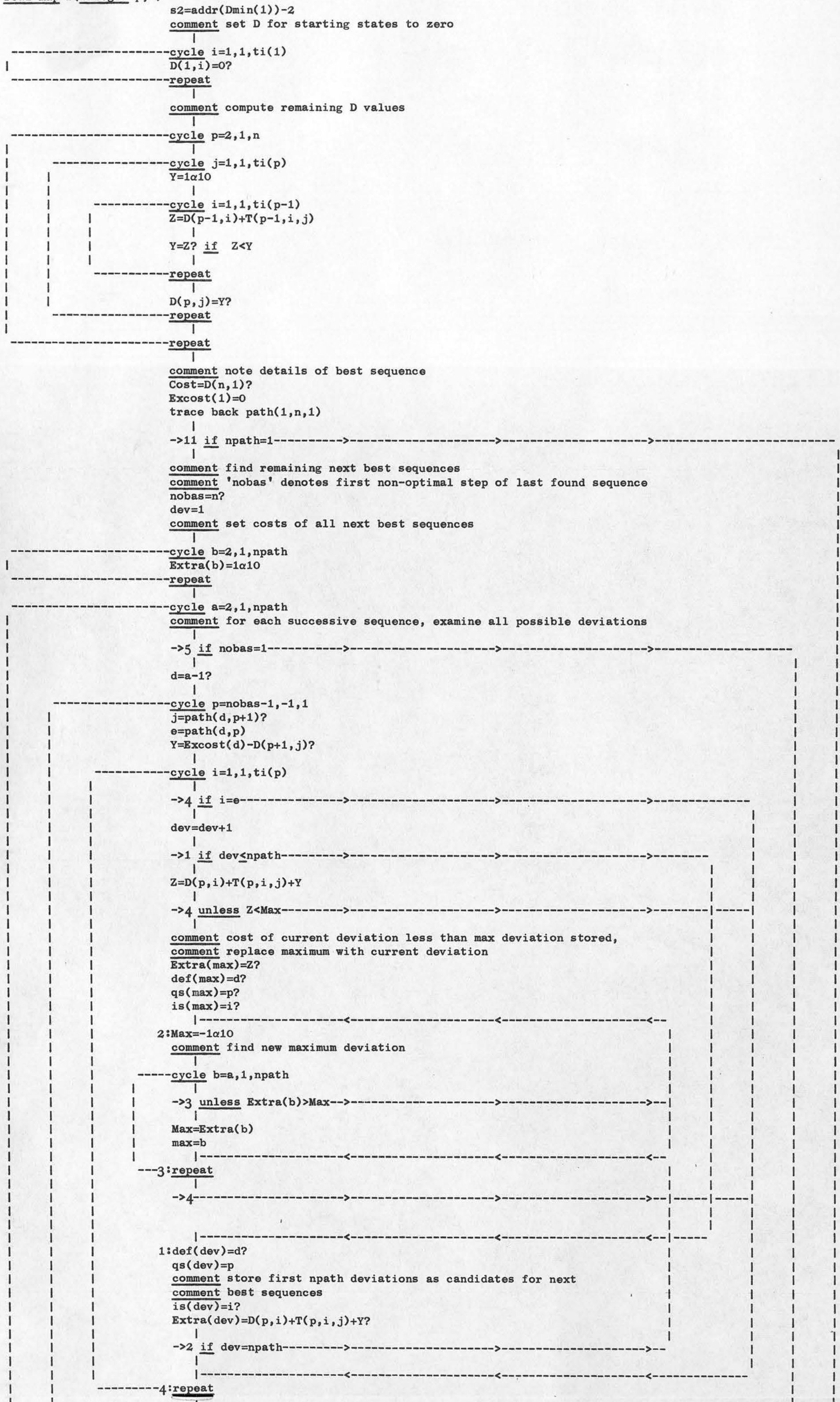
14(a) Flow-chart of dynamic programming analysis

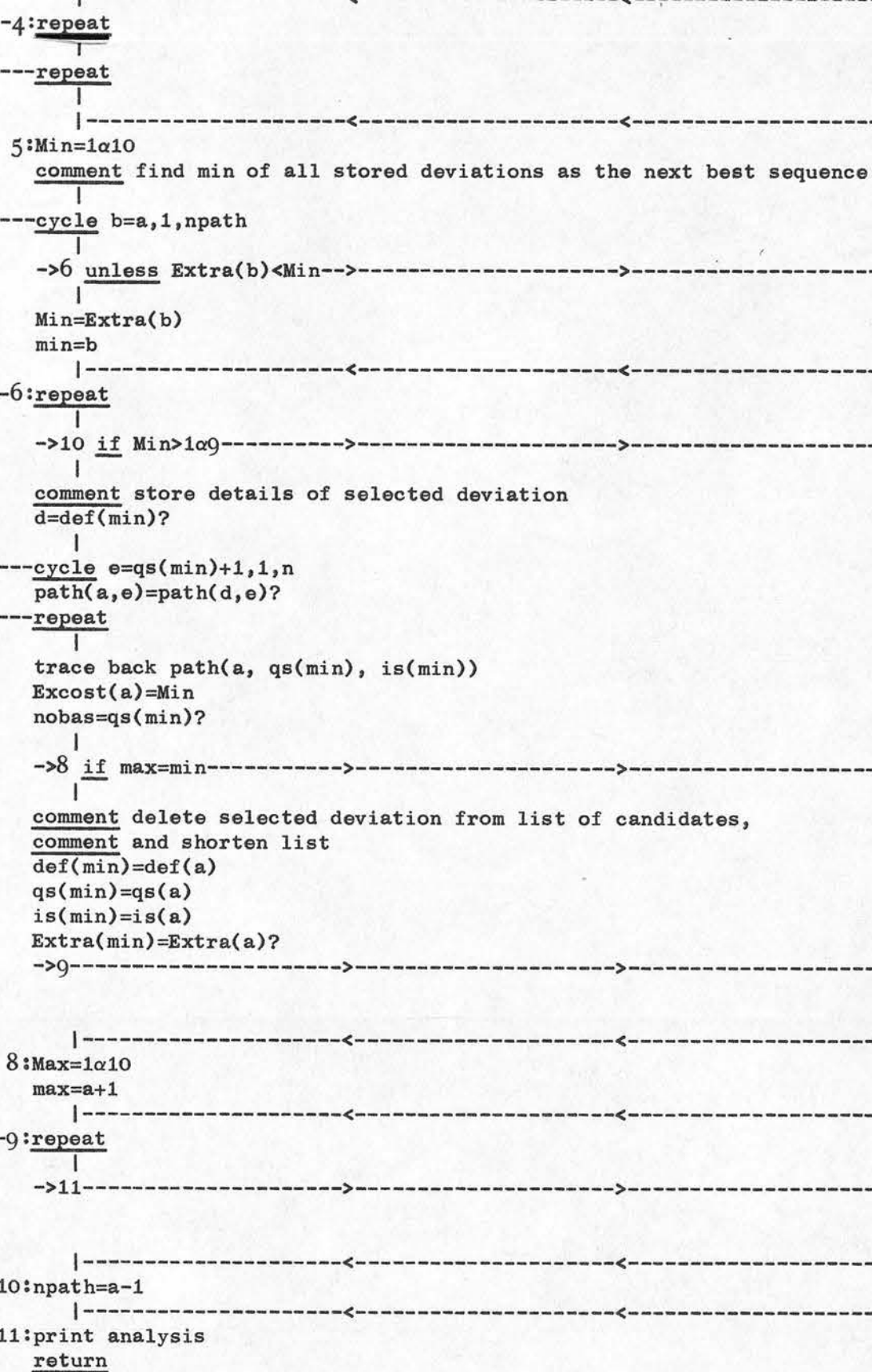
FLOW CHART OF
routine dp analysis

```

comment for n intervals, ti(p) states per interval, transition
comment costs T(p,i,j): routine finds the npath cheapest state
comment sequences, places min cost in Cost, extra cost for
comment subsequent alternatives in Excost, and details of sequences
comment in path
integer p,i,j,a,b,e,d,dev,max,min,nobas,s2
real Y, Max, Min, Cost, Z
array Dmin(1:td(n)+ti(n)), Excost(1:npath), Extra(2:npath)
integer array path(1:npath,1:n), def, qs, is(2:npath)
routine spec trace backpath(integer q,r,s)
routine spec print analysis
real map D(integer p,i)

```

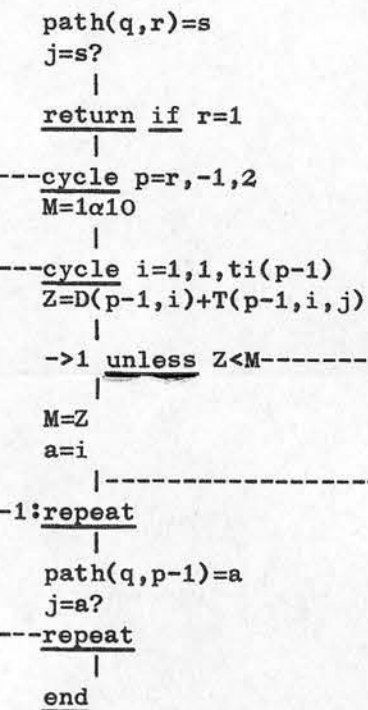




END OF FLOW CHART

FLOW CHART OF
routine trace back path(integer q,r,s)

comment subroutine to trace best sequence from state s, interval r,
comment back to starting interval, results stored as first part of qth
comment next best sequence
integer j,p,i,a
real M,Z



END OF FLOW CHART

14(b) Computer-prepared flow-chart

Rating MVA	Percent ^{unit} Impedance	Total Cost	Installation Cost	Removal Cost
5	0.0058	6000	300	300
7.5	0.0075	7500	350	350
10	0.01	9000	450	450
15	0.01	16000	800	800

15(a) Transformer details

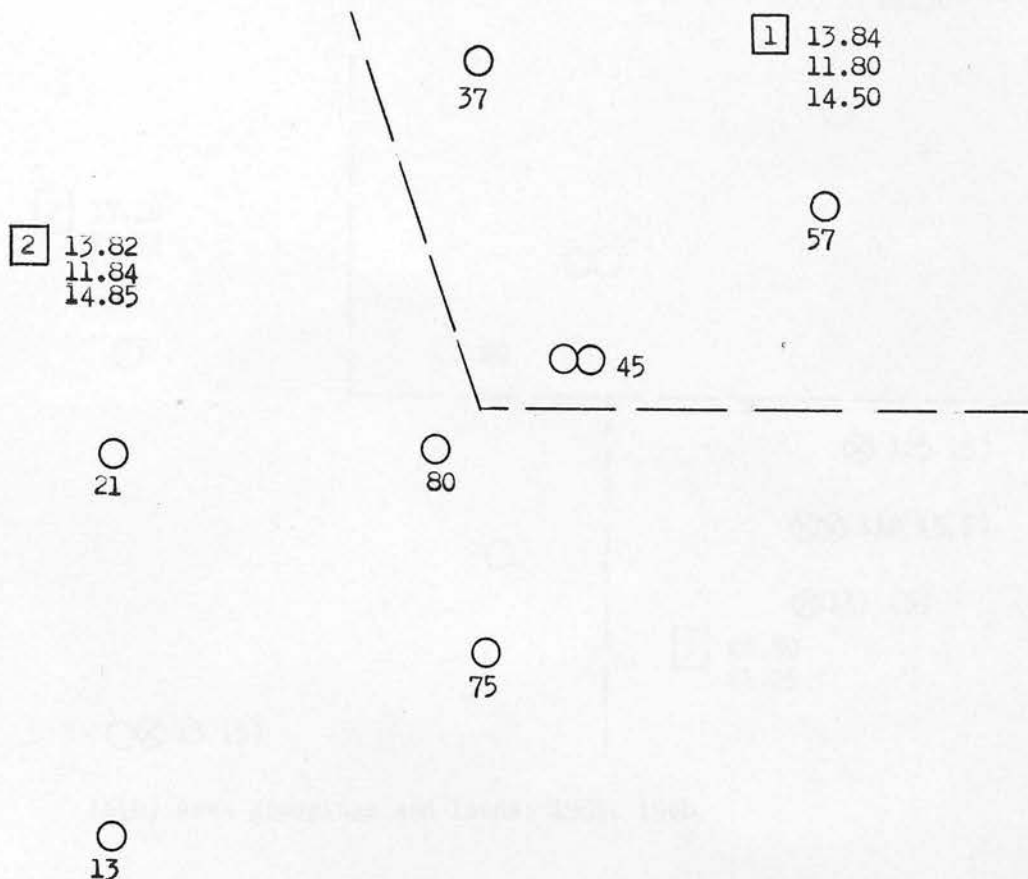
For use in:-	Total Cost	Installation Cost	Removal Cost
Spur unit	6000	300	300
Ring-main unit, with isolators and protective devices.	10000	500	500

15(b) Switchgear details

Size sq.ins.	Rating MVA	Impedance ohms/1000yd.	Cost per yd.	Laying Charge
0.3	20	1.24	9	2

15(c) Cable details

Transformer and switchgear input data, and cable data
Figure 15



16(a) Area groupings and loads: 1956, 1957, 1958

Key:-

1 29.75

Area no. and load (MVA)

○

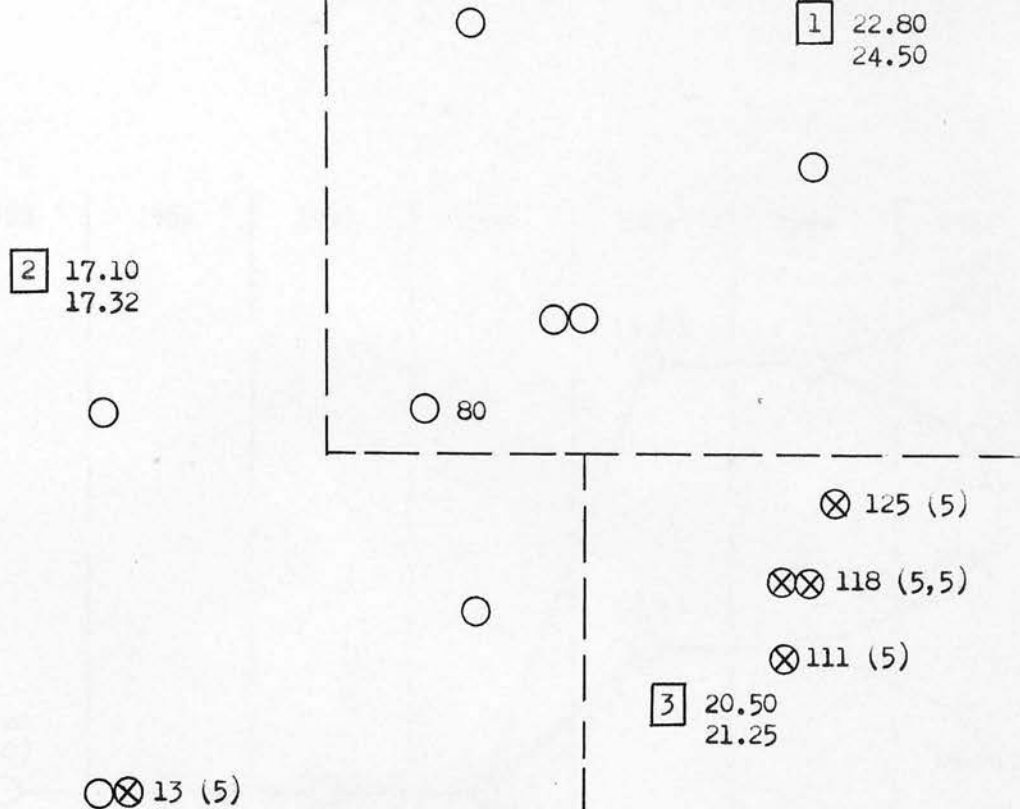
Existing transformer

○ 81

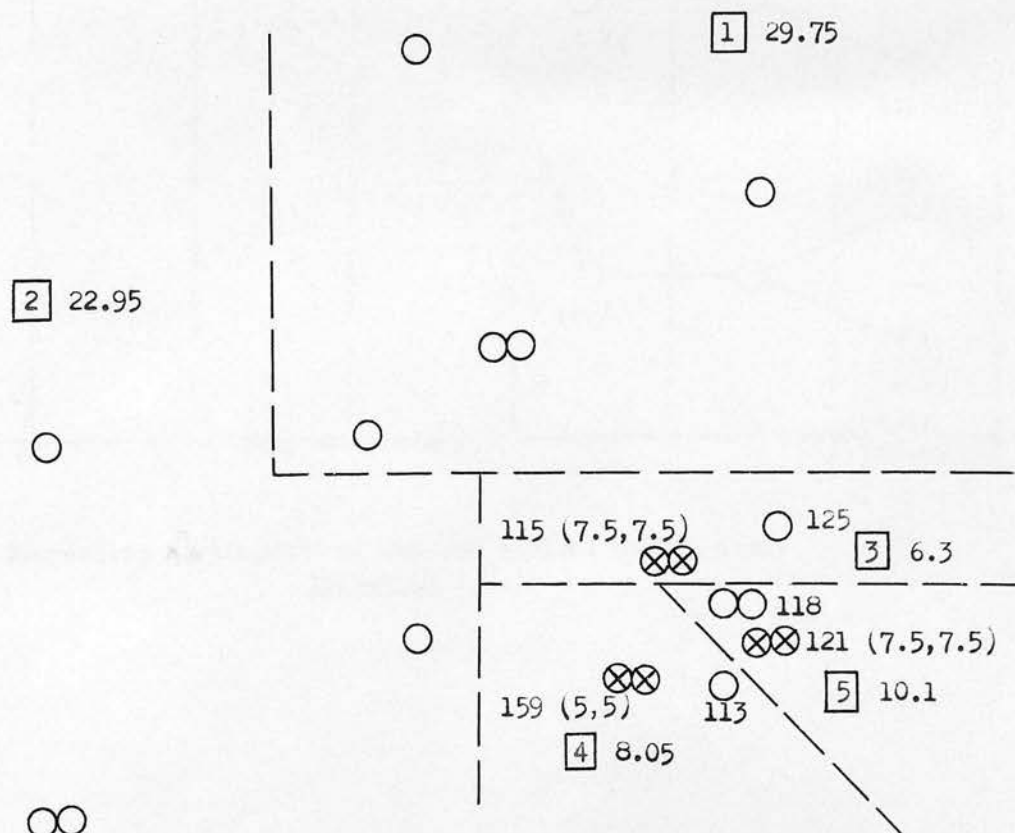
" " transferred from another area,
with map node no.

⊗ 113 (5)

New transformer with map node no. and MVA rating.



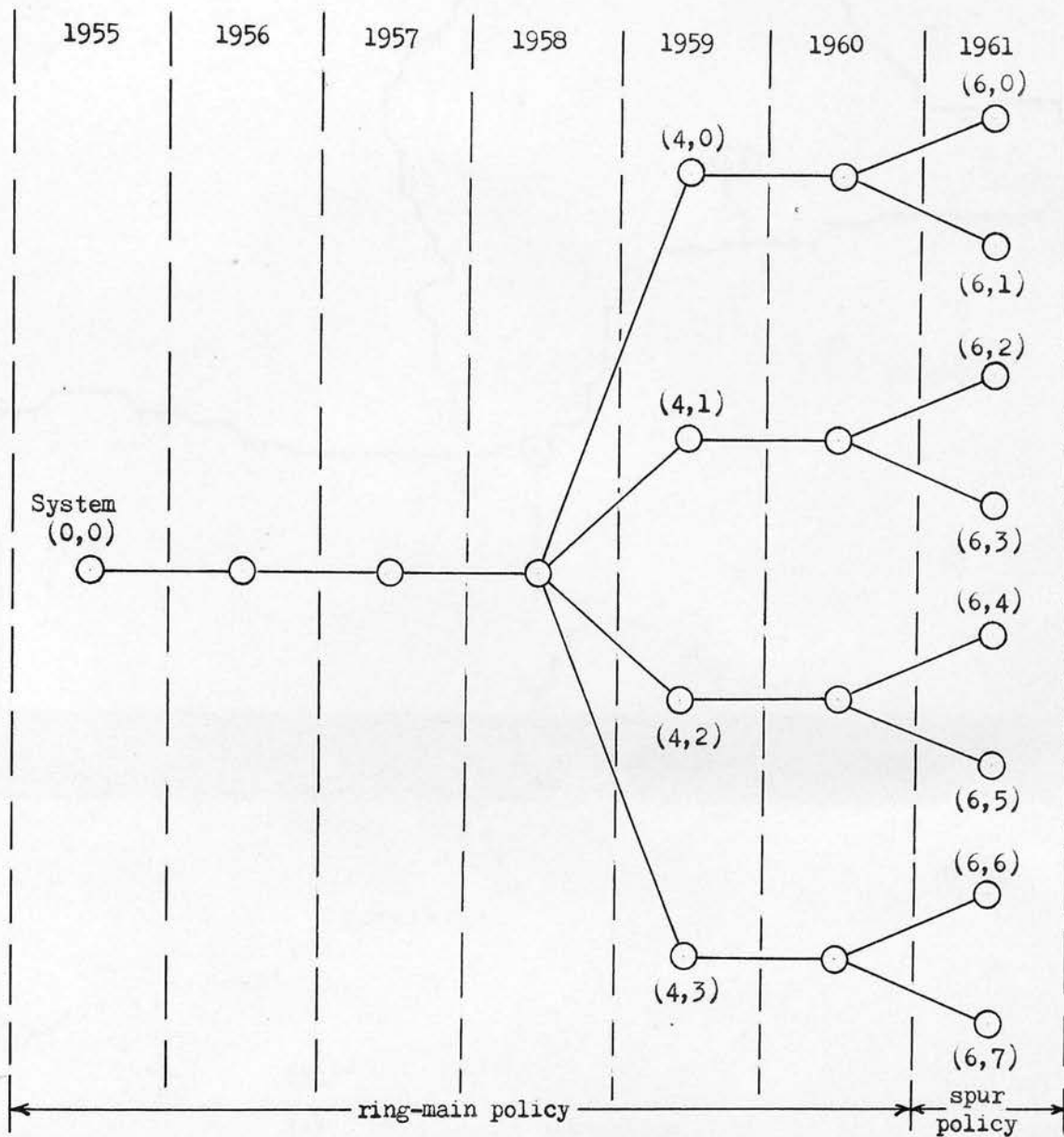
16(b) Area groupings and loads: 1959, 1960



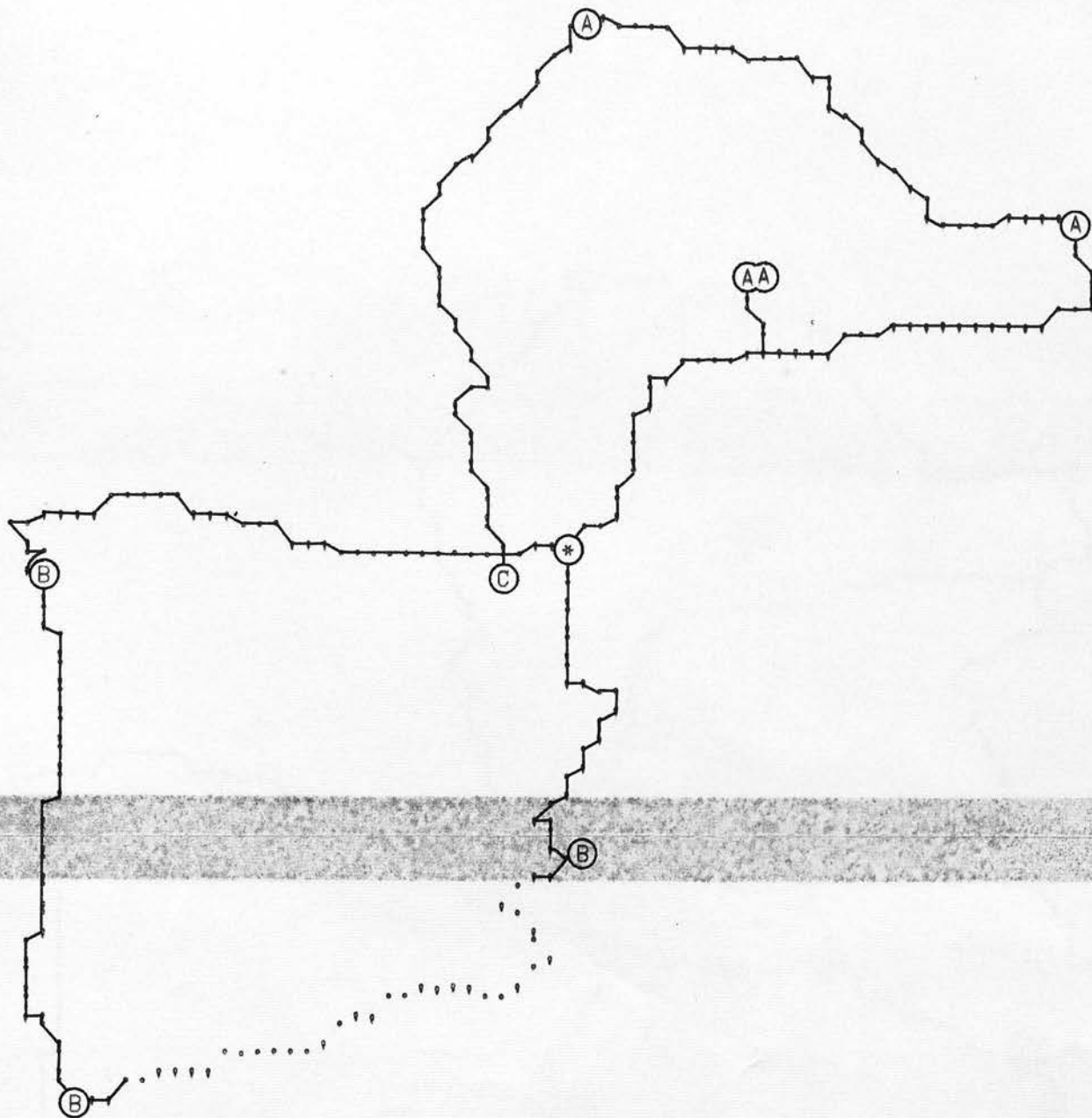
16(c) Area groupings and loads: 1961

Load data for program - transformer positions and area groupings

Figure 16



Successive development of designs - first design study
Figure 17

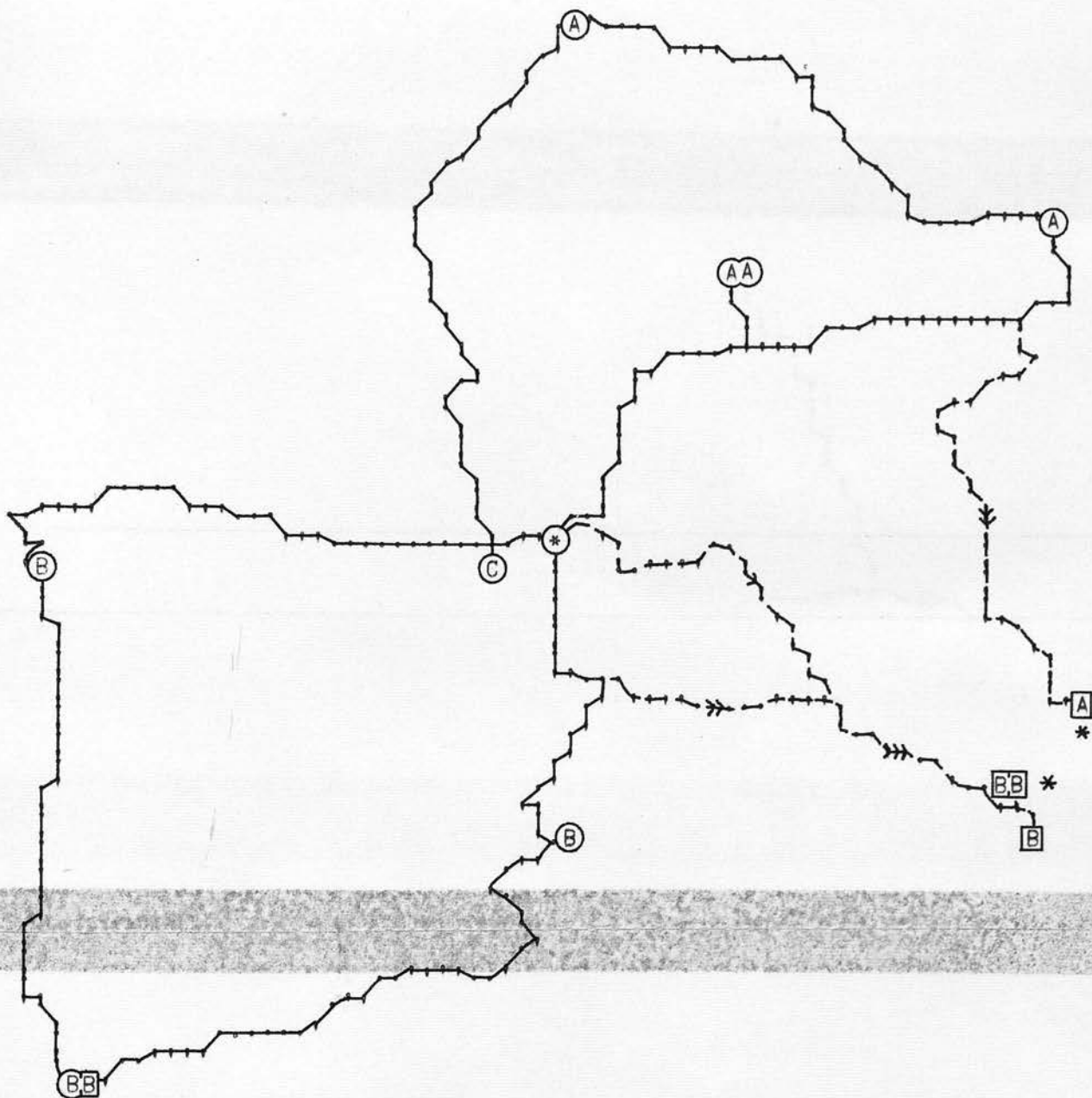


Key:-

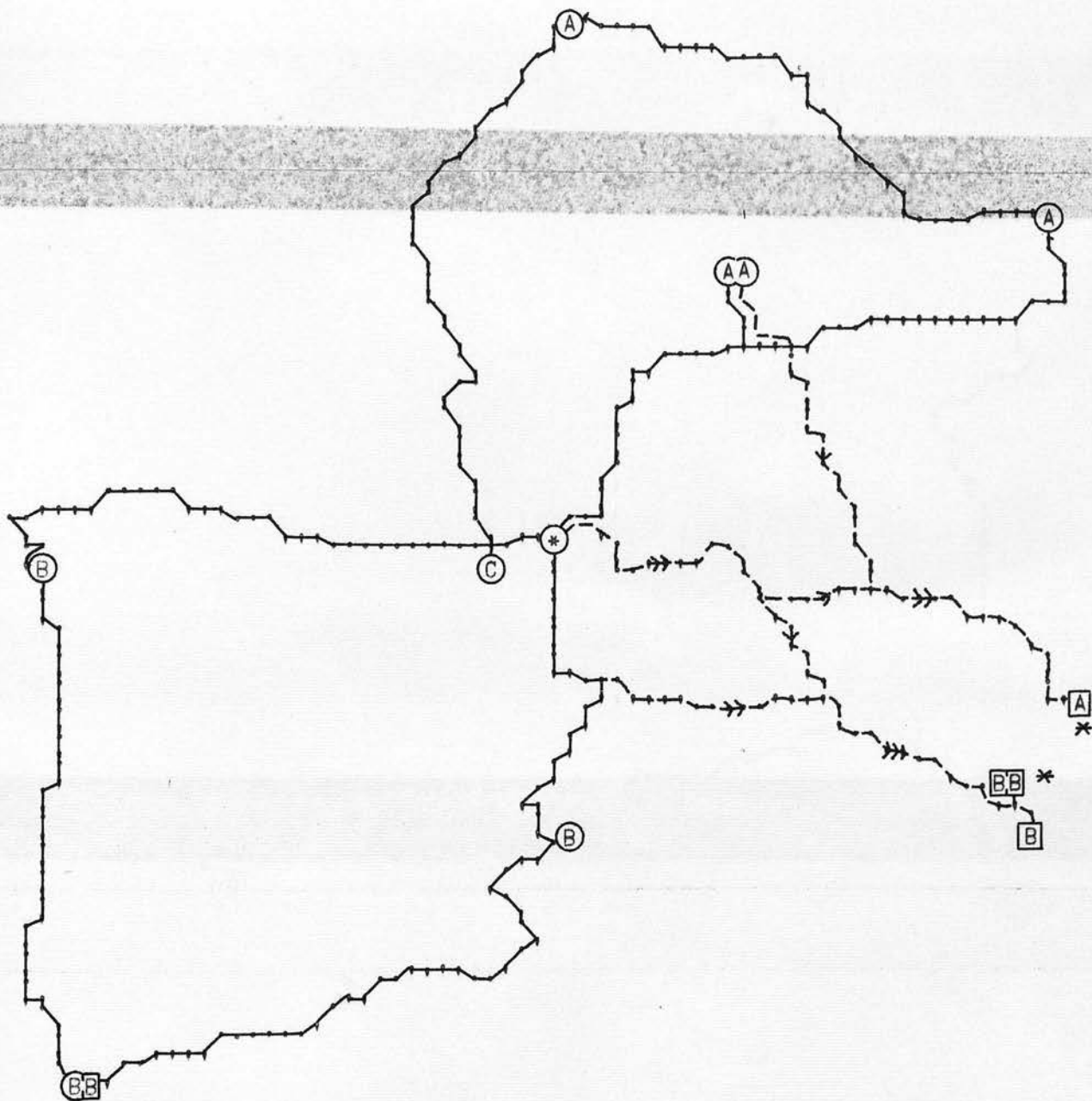
- (*) Grid supply substation
- (A) Load transformer
- [A] New load transformer
- (A)* Load transformer increased in size

———— Cable route

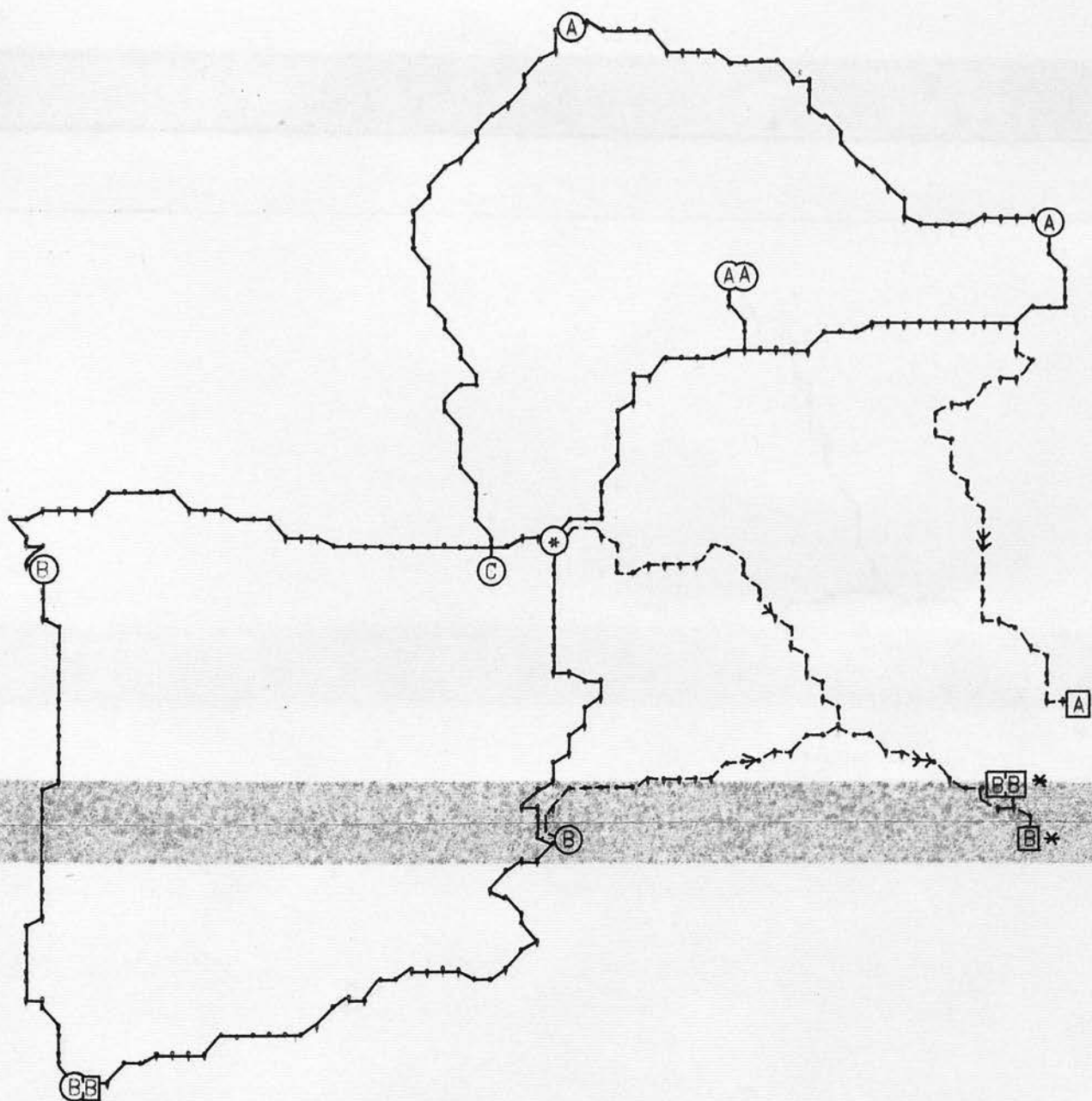
—>>>— New cable route (no. of arrows = no. cables)



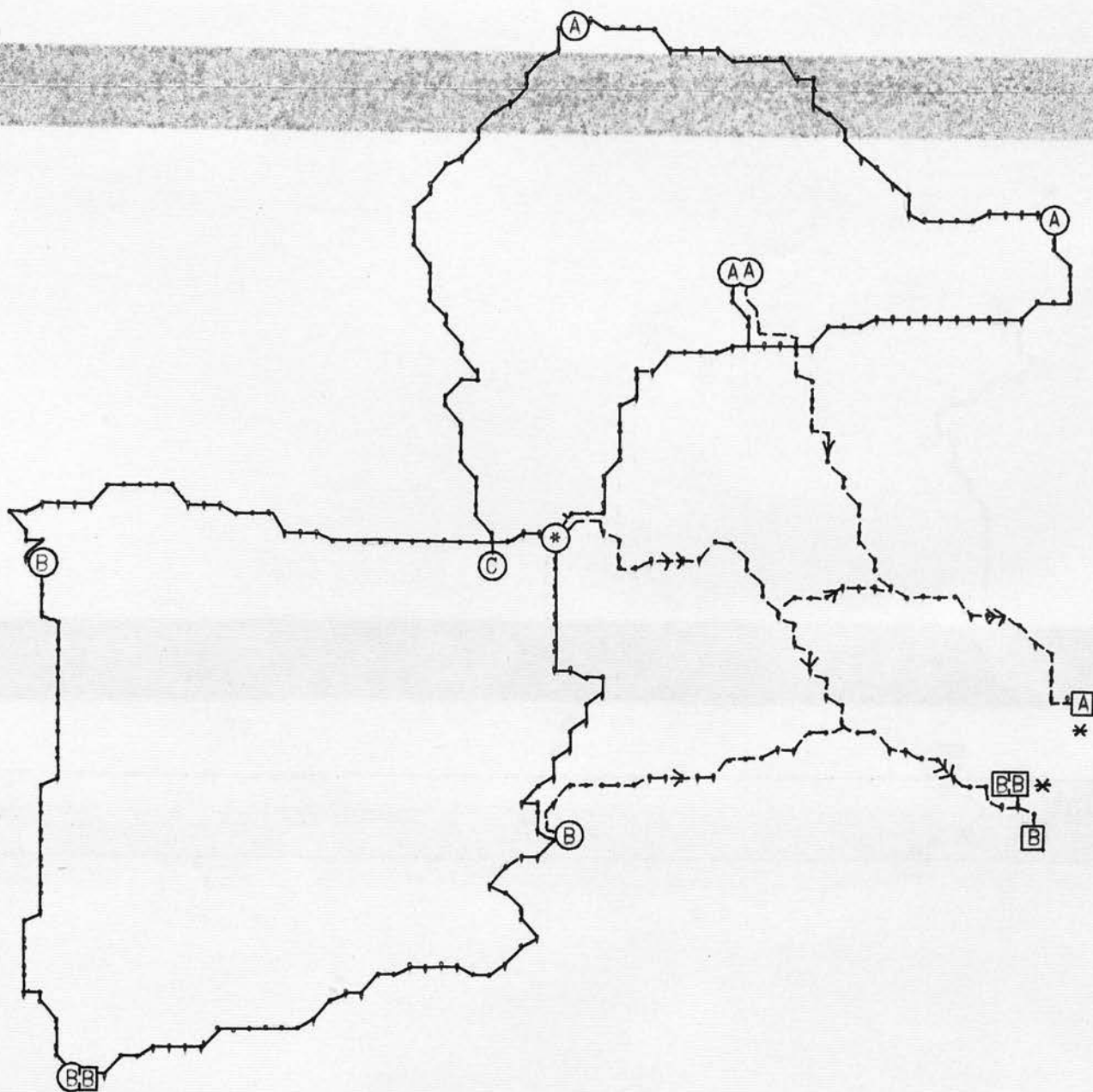
18(b) System (4,0)



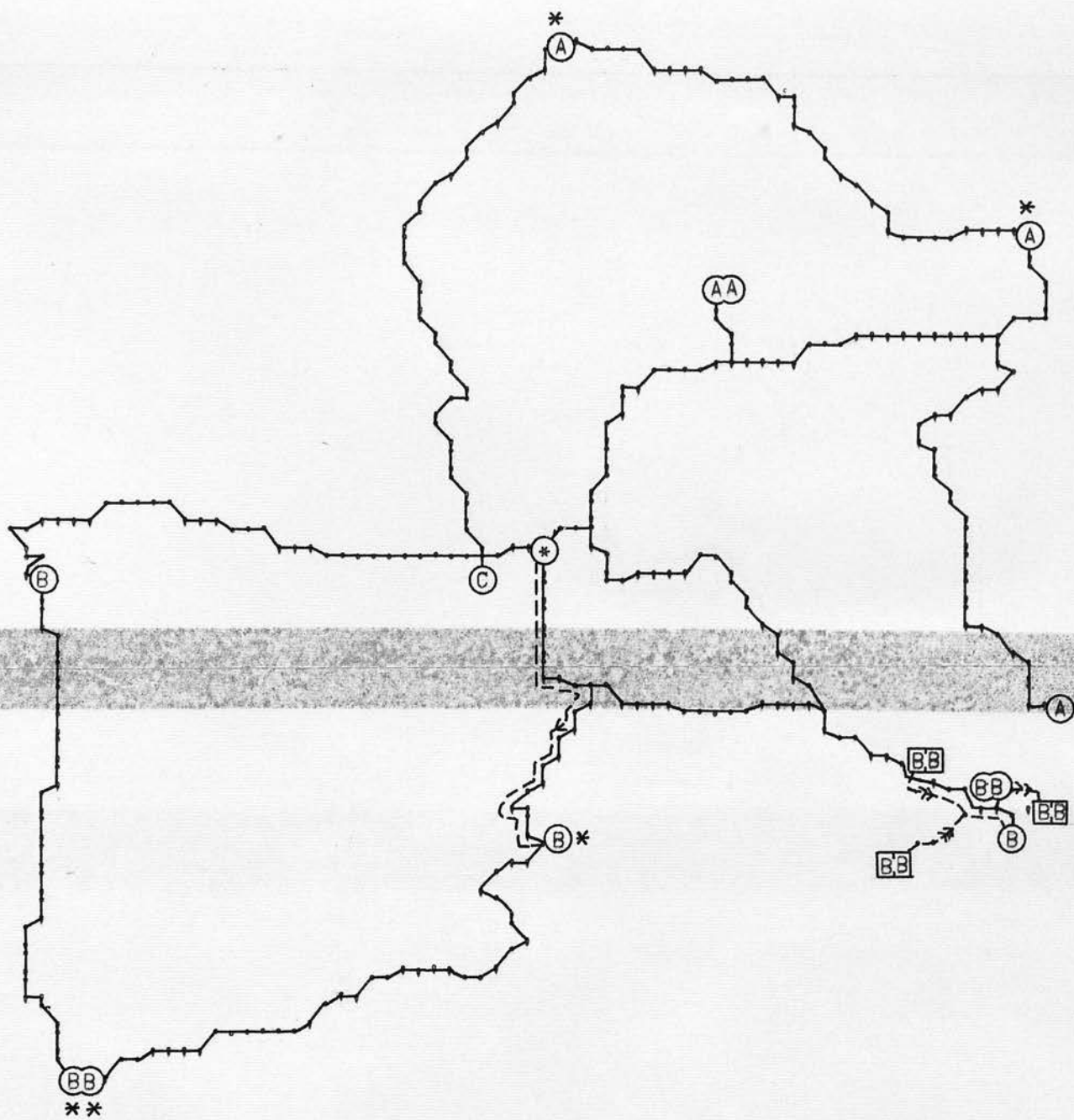
18(c) System (4,1)



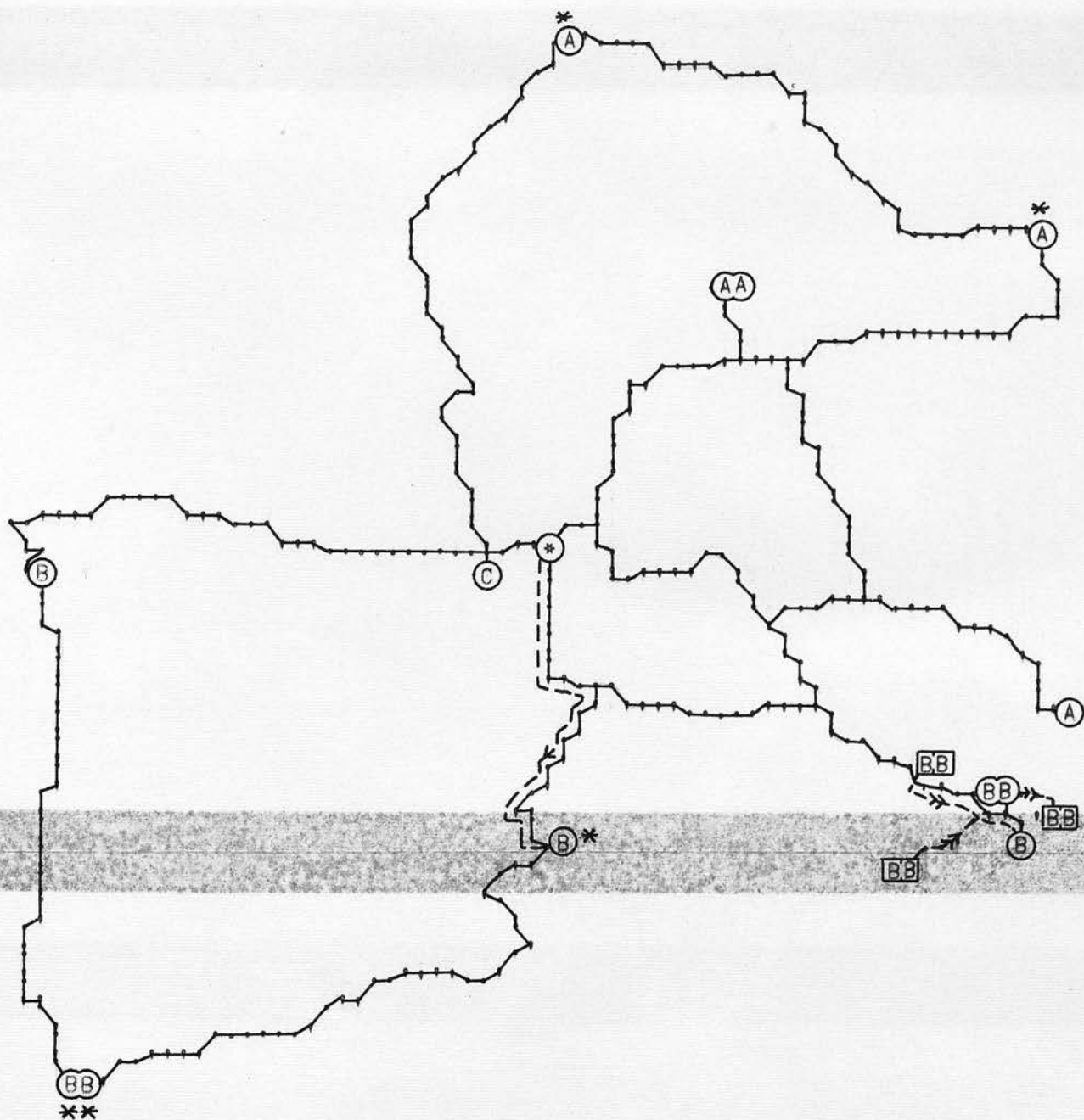
18(d) System (4,2)



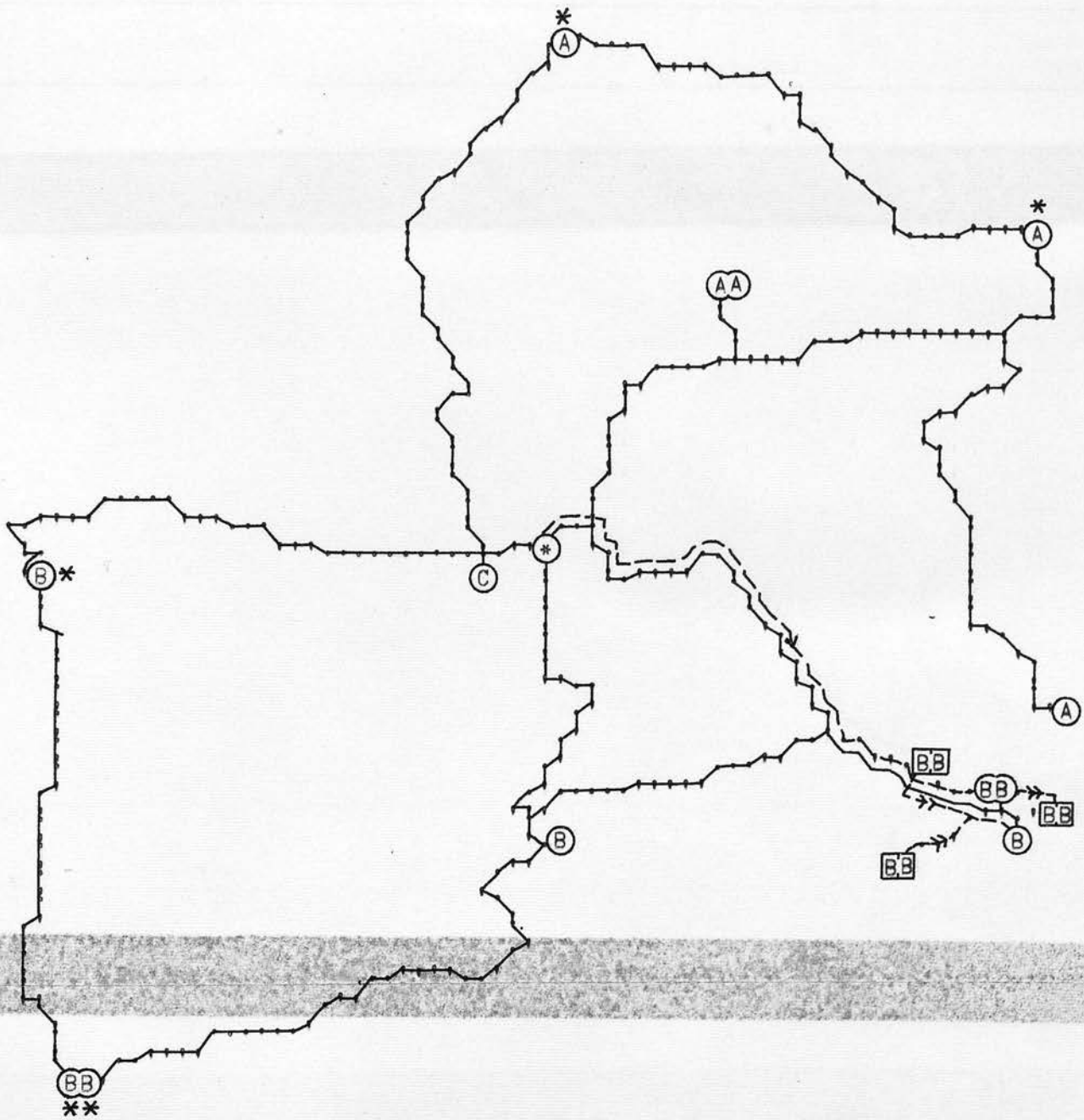
18(e) System (4,3)



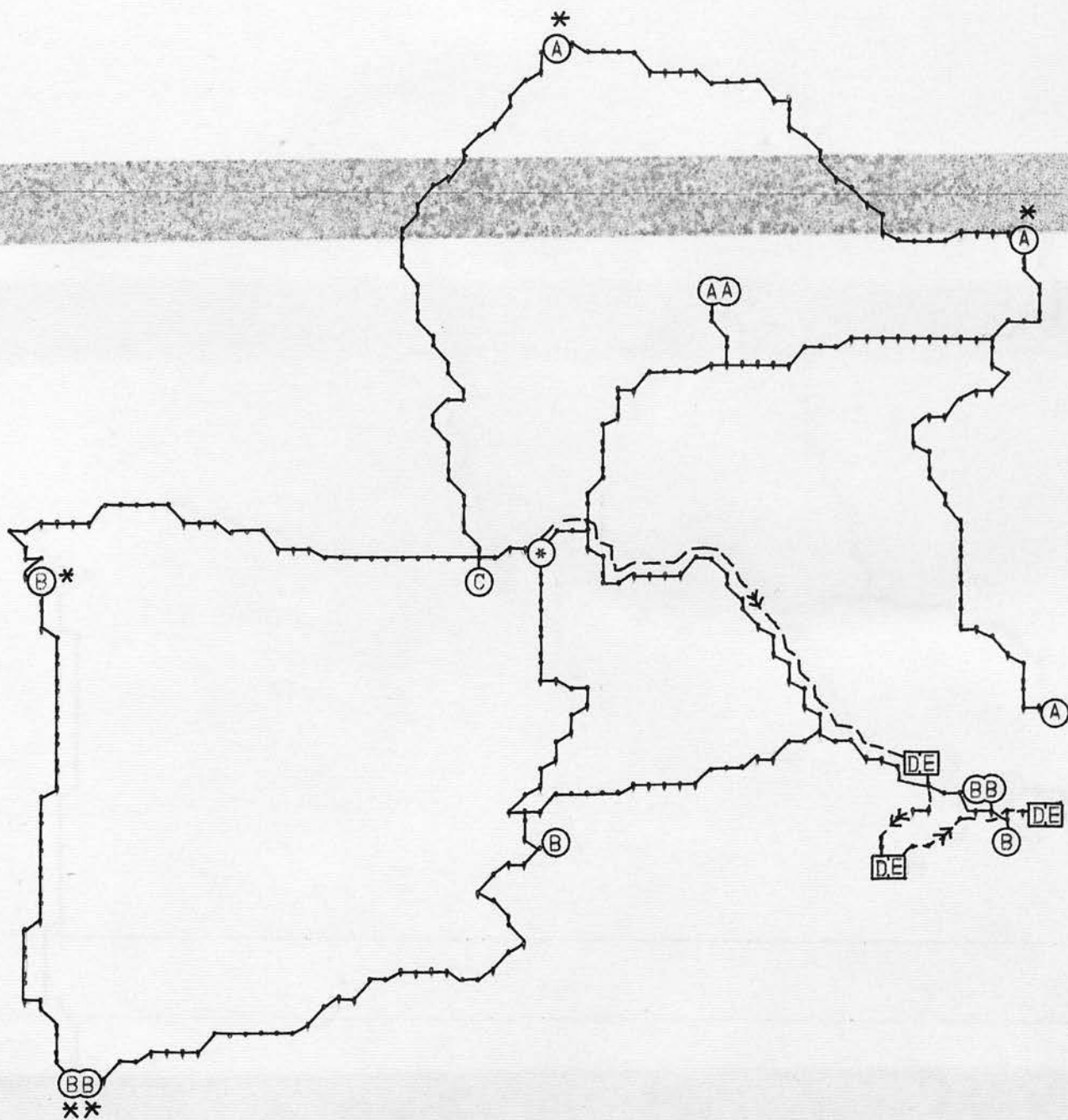
18(f) System (6,0)



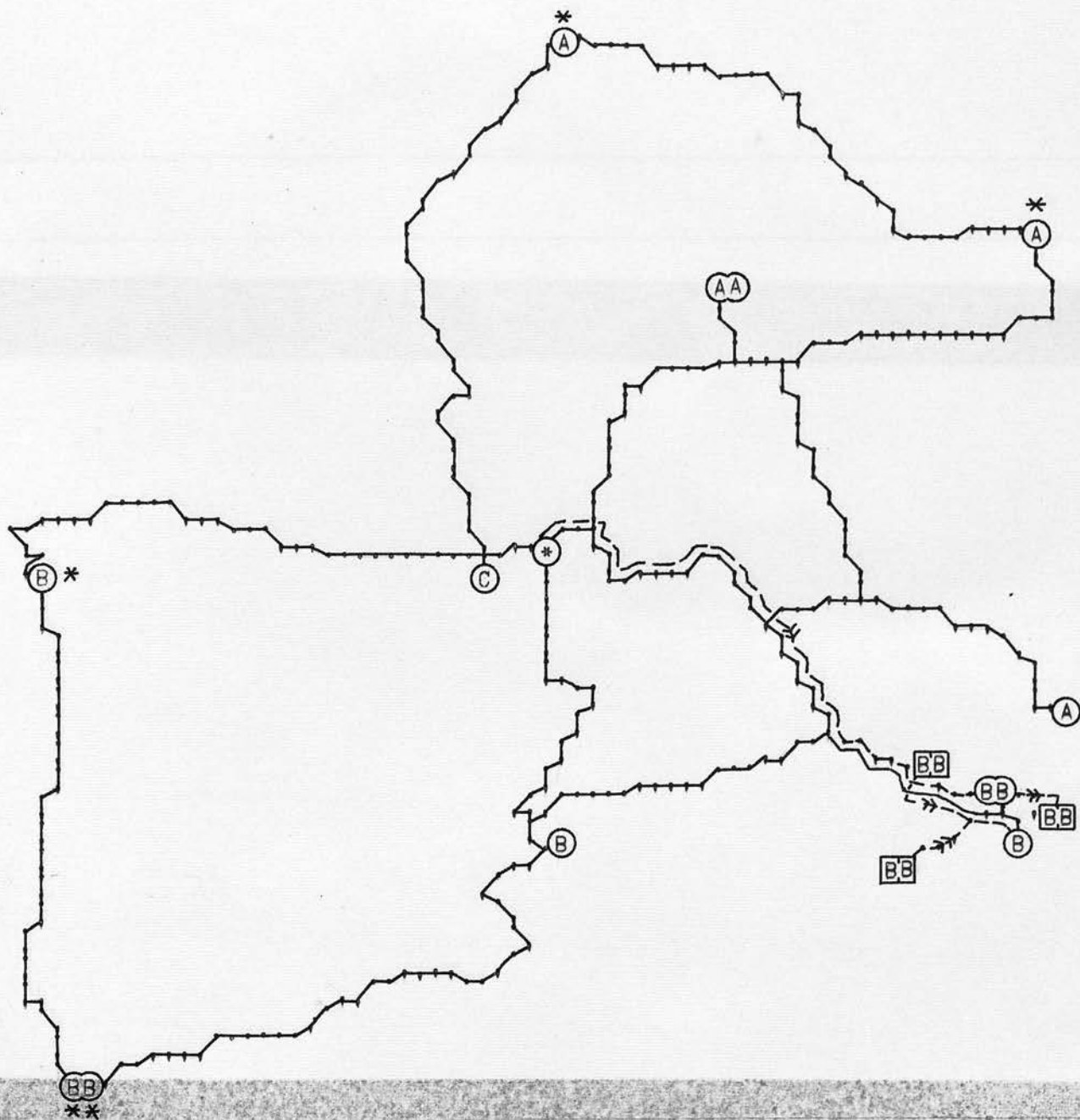
18(h) System (6,2)



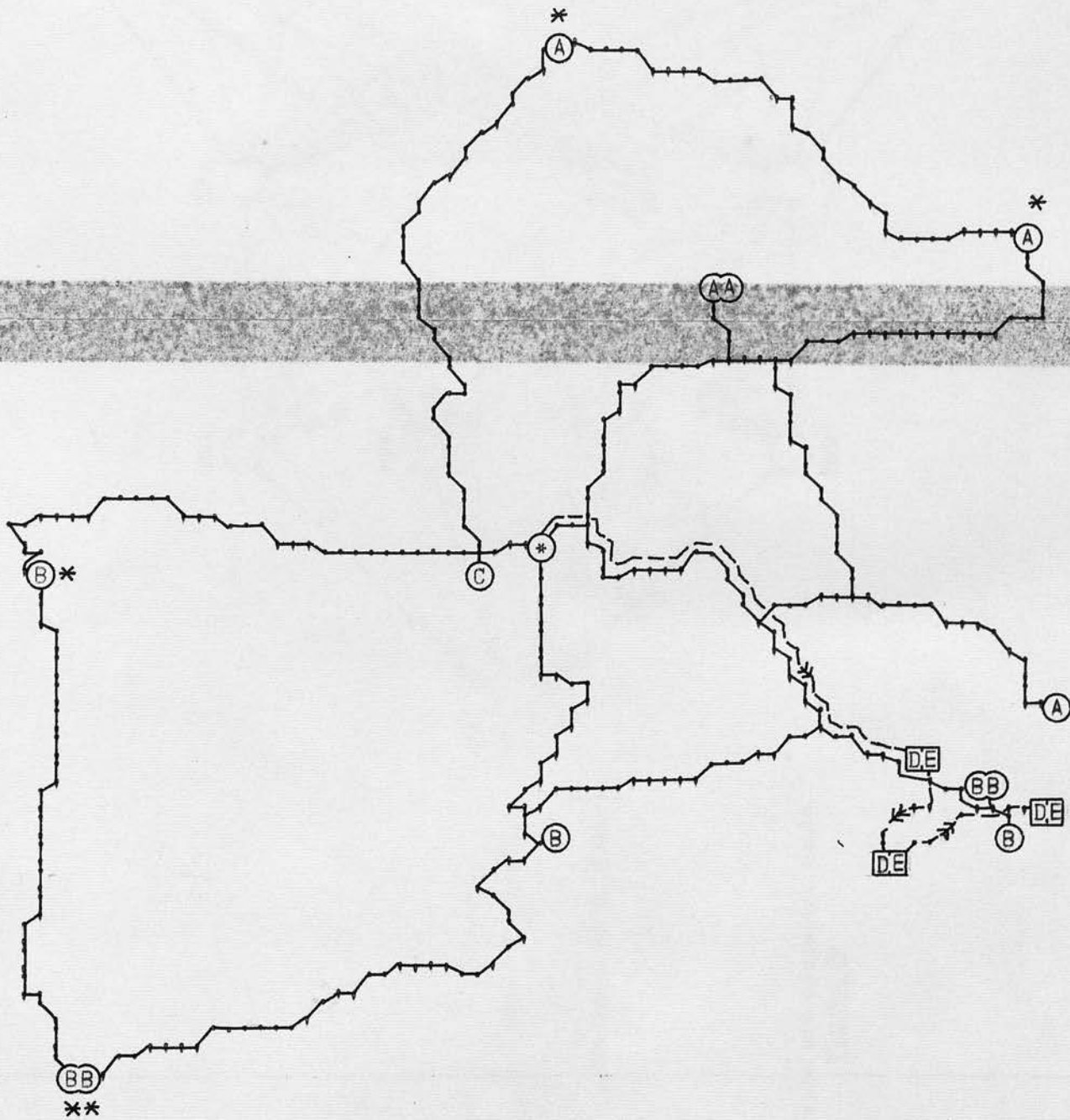
18(j) System (6,4)



18(k) System (6,5)

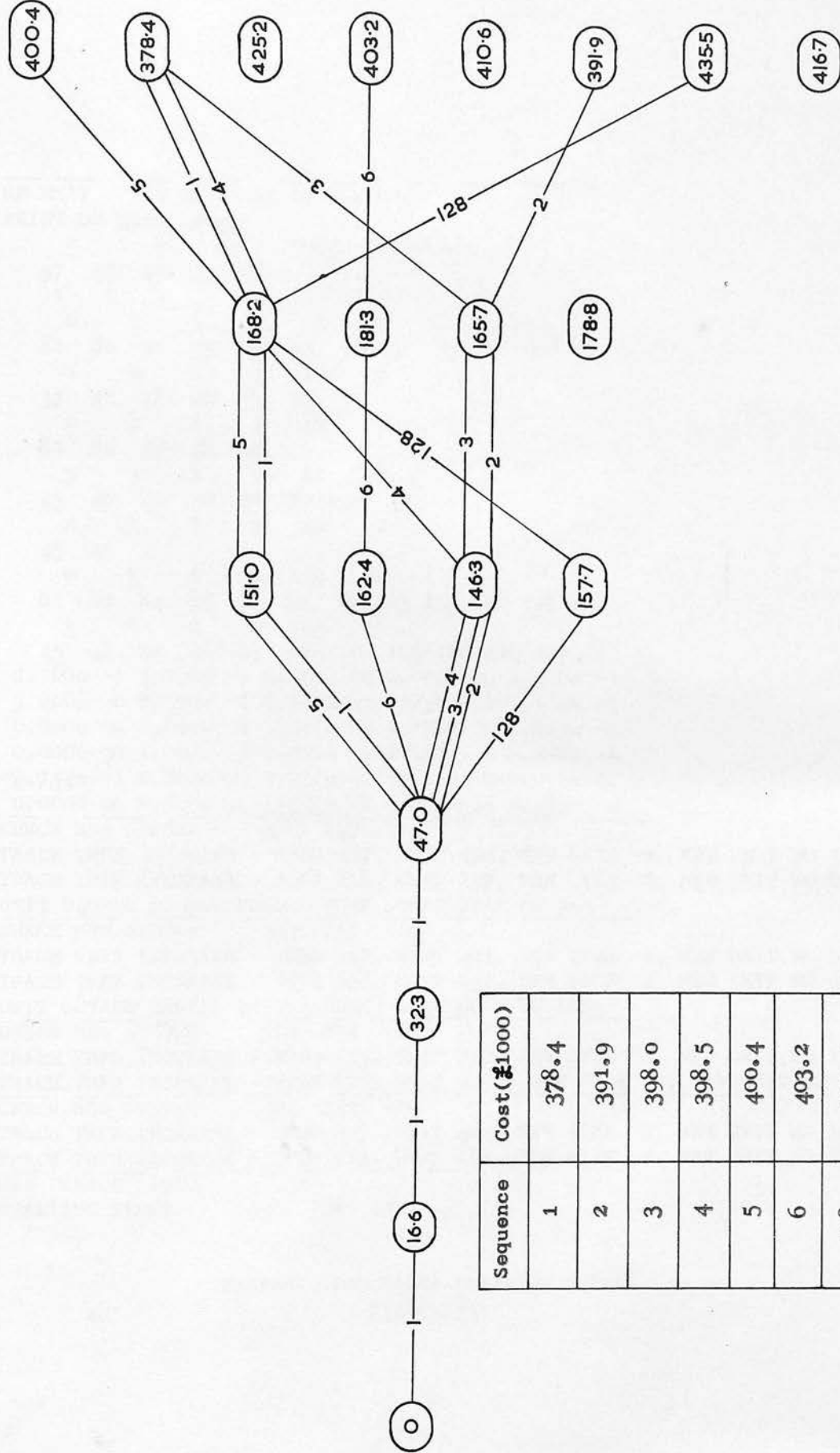


18(1) System (6,6)



18(m) System (6,7)

Set of designs produced in first design study
Figure 18

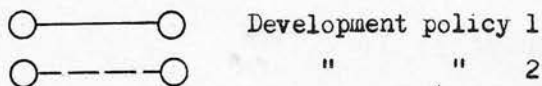
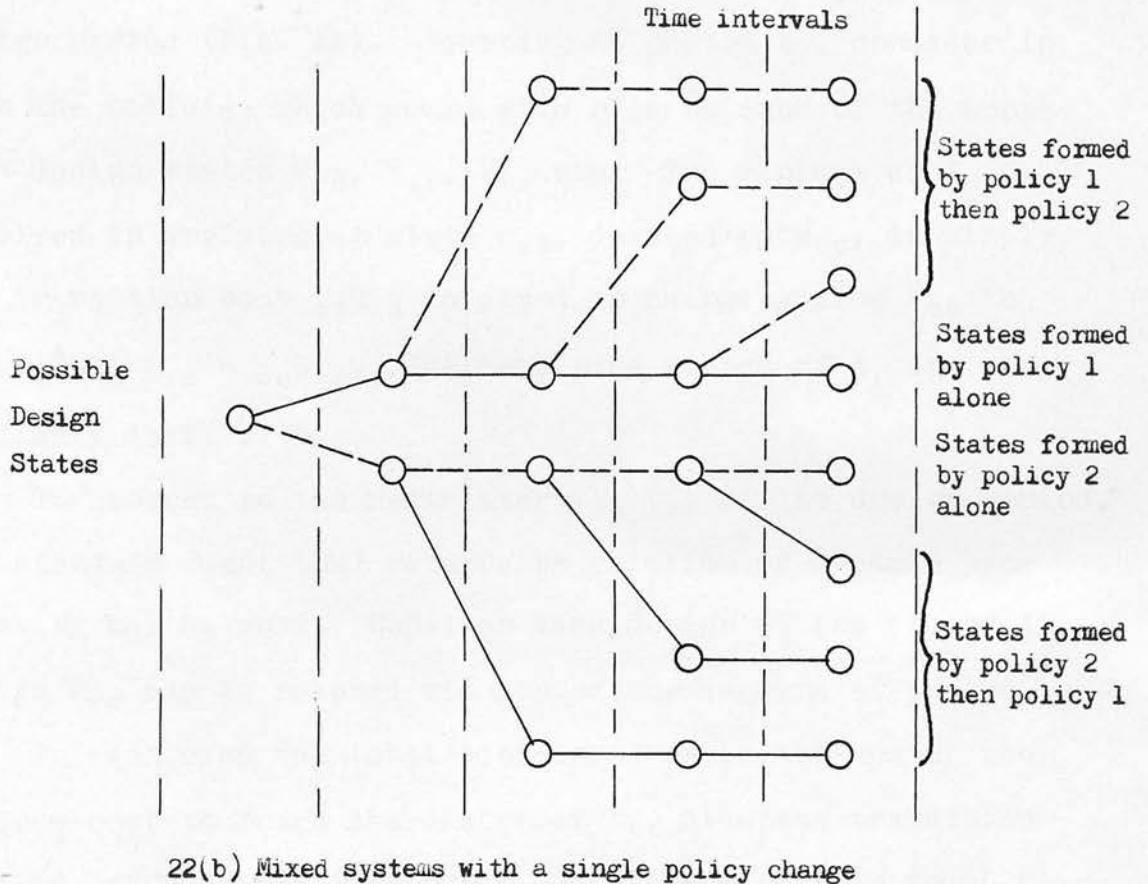
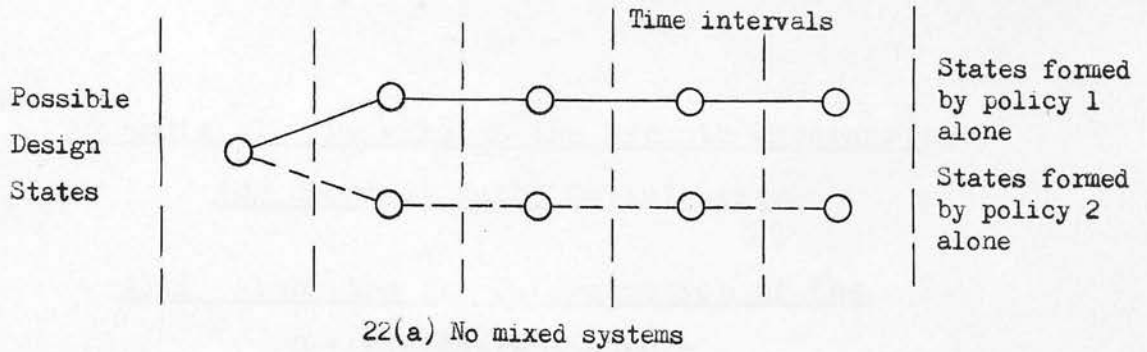


Sequence	Cost (£1000)
1	378.4
2	391.9
3	398.0
4	398.5
5	400.4
6	403.2
128	520.4

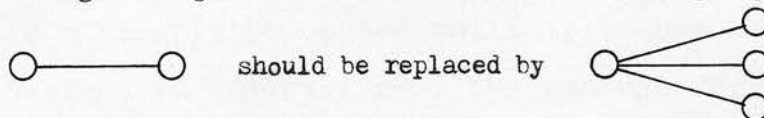
Dynamic programming and best sequences
Figure 20

RM UNIT	1	REPLACED BY NEW UNIT		
PRINT OF UNIT	464			
5	1	81	91	13023
37	45	45	57	127
1	1	1	1	1
0	1	1	0	1 9
81	80	32	33	34 35 38 151 37
1	4	2	2	10 6
37	41	43	46	53 57
0	2	1	2	16 5
81	82	47	48	45
3	4	2	2	21 8
45	48	49	50	51 52 156 57
2	3	2	2	29 2
45	45			
0	5	1	1	152 12
81	82	84	83	94 95 96 103 105 106 126 127
3	5	2	0	164 11
45	48	49	98	97 96 103 105 106 126 127
2.169 α	-1	4.046 α	-1	0.000 α -99 0.000 α -99 2.418 α -1
3.006 α	0	6.579 α	-1	6.582 α -1 1.738 α 0 3.380 α -1
0.000 α -99	1.621 α	0	1.621 α	0 1.178 α 0 8.325 α -1
0.000 α -99	-1.000 α	3	1.622 α	0 1.178 α 0 8.328 α -1
-2.744 α	-1	0.000 α -99	-3.221 α	-1 3.112 α 0 6.052 α -1
0.000 α -99	0.000 α -99	-2.553 α	-1	0.000 α -99 2.439 α 0
CHECK NEW SYSTEM	379	137	119	
TRACH TRFR INCREASE	-	NODE 127, UNIT 379, NEW SIZE	2, NEW UNIT NO 555	
TRACH TRFR INCREASE	-	NODE 118, UNIT 137, NEW SIZE	2, NEW UNIT NO 640	
UNIT 640	ADD RADIAL FEEDER NODE	6NEW UNIT NO 755		
CHECK NEW SYSTEM	464	137	119	
TRACH TRFR INCREASE	-	NODE 127, UNIT 464, NEW SIZE	2, NEW UNIT NO 876	
TRACH TRFR INCREASE	-	NODE 118, UNIT 137, NEW SIZE	2, NEW UNIT NO 967	
UNIT 967	ADD RADIAL FEEDER NODE	6NEW UNIT NO 1082		
CHECK NEW SYSTEM	379	252	119	
TRACH TRFR INCREASE	-	NODE 113, UNIT 252, NEW SIZE	2, NEW UNIT NO 1203	
TRACH TRFR INCREASE	-	NODE 118, UNIT 1203, NEW SIZE	2, NEW UNIT NO 1330	
CHECK NEW SYSTEM	464	252	119	
TRACH TRFR INCREASE	-	NODE 127, UNIT 464, NEW SIZE	2, NEW UNIT NO 1457	
TRACH TRFR INCREASE	-	NODE 113, UNIT 252, NEW SIZE	2, NEW UNIT NO 1548	
NEW PERIOD	1960			
STARTING STATE	555	755	119	

Extract from first computer output
Figure 21



Note:- In a development program where a number of alternative new designs are put forward at each step, each single growth



Stages in the use of mixed policies for ad-hoc design
Figure 22

Appendix A1. Details of the Dynamic Programming
(or Shortest Path) Optimisation

A1.1 Algorithm for Determination of the
Optimal State Sequence

Let $t_0, t_1, \text{-----}, t_k$ represent the intervals of the design period (Fig. A1). Starting at period t_1 , consider in turn the policies which would give rise to each of the possible design states P_{10}, P_{11}, P_{12} etc. The minimum cost involved in arriving at state P_{10} , denoted by m_{10} , is simply the transition cost ${}_{00}C_{10}$ involved in changing from P_{00} to P_{10} . Thus $m_{10} = {}_{00}C_{10}$. The remaining states of t_1 are similarly dealt with.

To proceed to the next interval, t_2 , of the design period, the standard functional recurrence relation of dynamic programming may be used. Consider each design of the period t_2 . Design P_{20} may be reached via any of the designs of period t_1 . In each case the total cost involved is the sum of the minimum cost to reach the design of t_1 , plus the transition cost to reach P_{20} in t_2 . Hence the minimum cost to reach P_{20}

$$m_{20} = \text{minimum of } (m_{10} + {}_{10}C_{20}) \\
(m_{11} + {}_{11}C_{20}) \\
(m_{12} + {}_{12}C_{20})$$

This is repeated for every design state in period t_2 . Each interval is similarly processed until t_k is reached. If there are r_{n-1} designs in interval $n-1$, the general formula becomes

$$m_{ni} = \min(m_{n-1,j} + {}_{n-1,j}C_{n,i}); \quad j = 0, 1, \text{-----}, r_{n-1}-1$$

Finally at interval t_k , the optimal overall cost is

given by the minimum of

$$m_{k,i}; \quad i = 0, 1, \text{-----}, r_{k-1}$$

Starting from the state at t_k corresponding to the minimum cost, the step from t_{k-1} which contributed to this cost may be found. Continuing in this manner, the complete sequence which gave rise to the optimal cost may be traced back to the starting state at t_0 .

The number of comparisons (between one number and the sum of two other numbers) which must be performed during the process may be taken as an indication of the quantity of computation required. Suppose there are r states in each of k intervals. Then for the forward process there are $r + r^2(k-1) + r$ comparisons, and for the backward process $r \times k$, giving $(r^2(k-1) + r(k+2))$ in all, or approximately $r^2 \times k$.

Al.2 Algorithms for the Nth-best Path

There are two possible methods of obtaining not only the best, but also the 2nd, 3rd, etc. best sequences:-

Al.2.1 Extended Dynamic Programming Method [27]

This is a logical extension of the method of section Al.1. Instead of computing at each stage merely the minimum cost m_{ni} to reach any particular design, the N best values m_{nia} ($a = 1, 2, \text{----}, N$) are computed and stored. In this case each single comparison in the basic method of Al.1 is replaced by N comparisons, giving approximately $r^2 \times k \times N$ comparisons in all.

Al.2.2 Method of Best Path Plus Alternatives [26, 28]

This method assumes that the minimum functions m_{ni} and the best sequence have been found, and then proceeds to generate the alternative sequences. (The term "path" rather than "design sequence" will be used for the remainder of this section, as it illustrates more clearly the relationship with the shortest path theory upon which it is based.) The following brief description is based on the detailed analysis given in [28]. The slight variations from the approach of [28] gave an algorithm which was easier to program, and also slightly faster in use in cases where the number of alternatives required was small.

Every path which is not a best path must be a deviation from the best path. (The possibility of different paths of equal length will be ignored. The analysis remains valid in the event of such an occurrence.) A link (i.e. a path between two successive points or states) is said to be non-optimal if it does not constitute part of the best path from the starting point to the furthest end point of the link. A path from the starting point to the finishing point is said to be a Qth order deviation, where Q is the number of non-optimal links in that path. The root of a path is defined as that part of the path up to, but not including, the first non-optimal link.

In order to find the second best path it is only necessary to find the shortest of all the first-order deviations (since this must be shorter than any 2nd or higher order deviation). The deviations of interest may be made up of (a) paths which

diverge from the best path or never coincide with it, or (b) paths which converge onto the best path and continue along it. Type (a) paths may be covered by investigating the shortest paths which end at all those states in t_k except the state on the best path. Their values (m_{k0}, m_{k2} in Fig. A2) are possible candidates to provide the 2nd best path, and are hence stored. (These figures may arise from paths of a high order of deviation, but must include the best 1st order deviation if it is of type (a).)

The type (b) deviations have then to be investigated. For interval t_{k-1} , consider those paths ending on all states of t_{k-1} except that actually on the best path, and then joining the best path at t_k . Let $P_{k-1, j}$ be the state in t_{k-1} on the best path, $P_{k-1, i}$ some other state in t_{k-1} , and P_{kb} the state on the best path in t_k . The increased cost due to the first order diversion via $P_{k-1, i}$ is

$$(m_{k-1, i} + C_{k-1, i, k, b}) - (m_{k, b})$$

Hence the total costs of the paths diverted via the other states in t_{k-1} may be computed and stored.

This is repeated for each interval until t_1 is reached. The cheapest of all the computed deviations must constitute the 2nd best path.

For the 3rd best path it is necessary to find the shortest of the 1st and 2nd order deviations (except of course the best and 2nd best paths). It is unnecessary to consider the 2nd order deviations derived from the remaining 1st order

candidates for the 2nd best path, since each 2nd order deviation must be longer than the 1st order deviation from which it is derived. It is only necessary to add to the list of 1st order deviations those 2nd order deviations based on the 2nd best path which has just been formed. This requires a repetition of the process described above for type (b) deviations, performed on the root of the 2nd best path. The shortest of the paths from the augmented list will form the 3rd best path. This process may then be repeated in a similar manner to find as many of the next-best sequences as are required.

For a case of k intervals and r design states per interval, the determination of each additional path requires $\frac{1}{3}(kr)$ further comparisons, involving all deviations from the previously determined path. (The factor $\frac{1}{3}$ is an allowance for the increased requirement of comparing the sum of three numbers with one other.) Thus N paths require $(N-1) \times \frac{1}{3}(kr)$ additional comparisons, giving $r^2k + [(N-1) \times \frac{1}{3}(kr)]$ in all. Since it is not necessary to compute deviations from a whole path, but only from the root of each path, this figure is an over-estimate. On the other hand, certain subsidiary computations involving searches of a list of length N have also been neglected.

Comparing this with the $r^2k + (N-1)r^2k$ comparisons required by the method of extended dynamic programming, it can be seen that the second method requires less computation provided

$$(N-1)^{4/3} rk < (N-1)r^2k$$

$$\text{i.e. } 4/3 < r$$

Since this condition was always satisfied, the second method was adopted for use in this study. The implementation of the algorithm is described in section 6.2.2.

Appendix A2. The "Laynewspur" Problem

A2.1 Simplified Representation

The "laynewspur" subroutine was written to generate solutions to the following problem - given the geographic location of a supply substation and of a number of load substations, how should the load substations be connected into spur units so that a minimum length of cable is used? (A spur unit was defined in section 5.2 as a group of substations connected in series, one end of the series being connected to the supply substation, the other unconnected.) This is a simplified form of the problem as it arises in practice. Some of the possible complications, and the modifications required to the method of solution will be given in sections A2.2 and A2.3

An initial attempt was made to transform the problem into a linear programming problem of the "trans-shipment" type, where a cable could be looked upon as a shipment of goods from one substation to another, the cost being proportional to cable length. However, certain factors prevented such a transformation:-

- (1) There is no clear distinction between "source" and "destination" for the substations at either end of a cable.
- (2) There appeared to be no way of preventing the formation of closed rings of substations.
- (3) Although most substations have two cables laid to them, substations at the ends of spurs, and the supply

substation are exceptions to this.

It was then realised that the fact that most substations require two cables connected to them would allow the problem to be transformed into one of the "travelling salesman" type. In this type of problem a group of cities and the complete set of interconnecting distances is given, and it is required to find that route which, whilst visiting every city once and only once, and returning to the starting point, traverses the shortest distance. The two cables entering and leaving a substation may be regarded as the path taken by the salesman to visit the substation.

There still remains the question of substations at the end of spurs with only one cable connected, and of the supply substation with any number of connected cables. Consider Figs. A4(a) and (b), where S is a supply substation, and A and B are two load substations. Suppose that the route $S \rightarrow A \rightarrow B$ is being evaluated for a travelling salesman. For the case shown in Fig. A4(a), having traversed $S \rightarrow A$, the remaining portion of the route is obviously $A \rightarrow B$ as indicated. This would also apply if a cable route for a spur was being evaluated. Consider Fig. A4(b) however. In the strict travelling salesman problem $A \rightarrow B$ is still the remaining portion of this route. But in the case of cable routes, since $S \rightarrow B$ is shorter than $A \rightarrow B$ the obvious course in this case is to return to S (zero cost, since no cable laid), and lay a new spur $S \rightarrow B$ (cost equivalent to distance $S \rightarrow B$).

from this city to each one of the remaining cities.

Repeat this taking every other city as a starting point. In the second stage compute the minimum journeys starting from each city and visiting each other group of two. This requires the use of the two-city journeys computed in the previous stage. This process is repeated stage by stage (i.e. from each city to each possible group of three, four, five etc.), until in the final stage all cities are included, and the best journey selected. The method is a direct application of the usual functional recurrence equation of dynamic programming (compare with section 3.2.1):-

$$f(a; x_1, x_2, \text{----}, x_n) = \min_i [(a \rightarrow x_i) + f(x_i; x_1, \text{----}, x_{i-1}, x_{i+1}, \text{----}, x_n)]$$

where $i = 1, 2, \text{----}, n$, and $f(a, \underline{x})$ denotes the minimum journey from (a) visiting all the cities of set \underline{x} .

Fig. A5(a) shows a simple problem involving 4 load substations and a single supply substation. The corresponding distance table is given in Fig. A5(b), with the corrected distances corresponding to a return to the supply substation marked by asterisks. Fig. A5(c) shows the stage by stage solution of the problem. (After the value of each minimum function (f) the sequence which gave rise to this value has been recorded. It is not essential to carry this information forward. An alternative method is to evaluate the minimum functions only, until the full solution has been found, then trace backwards the sequence which gave rise to this solution.)

The solution - a single spur - is shown in Fig. A5(a).

When the "laynewspur" subroutine was used within the body of the design algorithm, actual road routes were used, rather than the straight line paths shown in the examples of this appendix.

For the small numbers of substations usually involved in individual problems it would generally have been more efficient to solve the problem of laying new spurs by the direct evaluation of all possible alternatives, but it was thought to be useful to have a routine capable of dealing with extreme cases.

This simplified approach to the problem was the one used in the main design algorithm of the long-term design program. In practice the problem is complicated by the load carrying limitation on the conductor used to form the spur units. It is quite possible for the "laynewspur" subroutine to design spurs which would give rise to cable overloads with the given substation loadings. In the design program as written, this situation would then be corrected by the unit load checking routine (section 5.2.6), which would split an overloaded spur into two smaller units. But it does not follow that the resulting solution is the optimum of all the solutions with acceptable cable loadings. A subsequent investigation showed how the dynamic programming approach could be extended to determine the optimal solution when a maximum cable loading was also specified. This refinement was not incorporated in the design algorithm, but since this problem is of general

interest to the electrical supply industry, it will be discussed in some detail in the following sections.

A2.2 Cable Loading Restrictions

Various approaches were considered to the problem of laying new spurs when cable loading restrictions were also specified. The possibility of obtaining an optimal solution ignoring this factor, and then applying corrections to alleviate any overloads proved to be unfruitful. Another approach also started from the optimal solution to the simplified problem. If overloads were present in this solution then the second best solution could be determined, and so on with the succeeding next-best solutions, until the first alternative with no overloads was found. This was rejected as being computationally inefficient, and also difficult to program. Evidently what was required was a means of rejecting partsolutions which involved overloads as they arose, rather than waiting for complete solutions to be formed and then checking for overloads.

Consider a typical point in the solution of the simplified problem in which $f(A; BCD)$ is being evaluated - where function f is defined as in section A2.1.

$$f(A; BCD) = \min. \text{ of } \begin{aligned} &[(A \rightarrow B) + f(B; CD)] \\ &[(A \rightarrow C) + f(C; BD)] \\ &[(A \rightarrow D) + f(D; BC)] \end{aligned}$$

Now if cable loading is to be considered, then before $[(A \rightarrow B) + f(B; CD)]$ is evaluated, it must first be decided whether substation A can be connected to the sequence (B; CD)

without causing an overload. This decision requires a knowledge of the actual sequence of (B; CD) which corresponds to $f(B; CD)$ - i.e. the history of how $f(B; CD)$ was arrived at has to be known. As described in section 3.2, the use of a cost function involving system history would invalidate the dynamic programming procedure.

In section 3.2.2 a discussion is given of how such a situation usually arises because the vector describing the system contains insufficient information. Obviously what is required in the present case is the addition of another item of information to the description (B; CD) to denote the loading on the end of the sequence to which new substations will be added. For example, (B; CD 10) would denote a load of 10 units on the cable supplying the start of this substation sequence. Figs. A6(a) and (b) show examples of a sub-system (B; CD) with initial cable loads 10 and 4 respectively, dependent upon the sequence of (B; CD). Taking the case of (B; CD 10), if the load at A was specified as 5 say, then it is at once apparent that $(A \rightarrow B) + (B; CD)$ would give rise to a group (A; BCD) with an initial cable loading of 15 i.e. (A; BCD 15). Thus if a maximum cable load of less than 15 has been specified, then the sequence $(A \rightarrow B) + (B; CD)$ would immediately be rejected.

Thus a means has been found of rejecting within the optimisation process any solutions which would involve cable overloads. However, this has been obtained at the expense of a two-dimensional dynamic programming process - i.e. there

are now two elements in the state vector. This means that the number of possible states to be considered by the program is considerably increased. For example

$$f(B; CD\ 10) = 8.6$$

$$\text{and } f(B; CD\ 4) = 12.1$$

are both possible candidates for the next stage of the optimisation process. Although the latter involves a greater length of cable, the low cable loading might still enable a better overall solution to be obtained in combination with a heavily loaded substation, which could not combine directly with the shorter alternative because of the loading restriction. Thus an increase must be expected in the amount of computation required to obtain a solution. The size of the increase will depend upon the ratio of average substation loading to the maximum permitted cable load. Provided this ratio is large (i.e. only a few substations can be permitted on any one spur,) then the number of possible loading combinations for each sub-group will remain comparatively small.

The use of a single sub-group with different loading conditions gives rise to another variation from the solution of the simplified problem. In the simplified problem, path $A \rightarrow B$ was replaced by $A \rightarrow S \rightarrow B$ (S denoting the supply substation) where this was shorter. But when cable loading restrictions are included, paths $(A \rightarrow B) + (B; CD)$ and $(A \rightarrow S \rightarrow B) + (B; CD)$ must both be considered. Although the latter might involve a longer cable route, it will also give the solution with the minimum initial cable loading i.e. the load on A alone.

Fig. A7(a) shows the problem solved for the simplified case, but with substation loads and a maximum cable loading also specified. The distance table remains as for the earlier problem (Fig. A5(b)). The stage by stage solution is given in Fig. A7(b), and the result has been shown on Fig. A7(a). (For the sake of convenience, the figure denoting the cable loading on each sub-group has been given after the actual substation sequence which gave rise to that corresponding solution, rather than within the sub-group label as above.) It will be seen that the computation required was approximately twice that of the simplified case.

A2.3 Further Extensions of the Method

The previous section described a major extension of the dynamic programming approach to include a more realistic representation of the problem. Further minor extensions are also possible.

A2.3.1 Several Supply Points

Problems may arise in which a group of load substations is to be supplied from more than one supply substation. This factor may easily be included. The distance table is set up as if there were only a single supply point. For each load substation, the distance from this supply point is specified as the distance to the nearest of the actual supply substations. The problem is then solved in the usual way. When the final solution is obtained, a route such as $A \rightarrow S \rightarrow B$ will then be interpreted as $A \rightarrow$ supply point nearest $B \rightarrow B$.

An example is shown in Fig. A8(a). In this problem an additional supply substation (S2) has been added to the otherwise unchanged specification of the previous example. The new distance table is given in Fig. A8(b), and the resultant solution is shown on Fig. A8(a).

In this case no attempt is made to control the load sharing or maximum load of either supply point. This aspect could be included by increasing the dimension of the problem yet again, to include in the state vectors the total loads so far reached on each supply substation. But the increase in computation would probably be prohibitive for any but the smallest of problems.

A2.3.2 Cost of Switchgear and System Losses

In all the examples given above it was assumed that the only costs to be minimised were those of the system cables, and that these would be proportional to route lengths. However, there is no reason why other costs should not be included. In particular each separate spur will require a circuit-breaker at the supply substation. The cost of these circuit-breakers could be included by a corresponding increase in the cost of all routes from the load substations to the supply point.

Another important factor to be considered is the cost of the power losses within the transmission or distribution system itself. To include the effects of these, it is possible to express the equipment costs (cables and switchgear) in terms of equivalent annual instalments, and then to seek

to minimise the sum of the annual capital instalments plus the yearly costs of the system power losses. This aspect can also be included in the present approach to the problem.

A description is given in section A2.2 of how the minimum functions $f(B; CD)$ could be expanded to become $f(B; CD L)$, where L is the corresponding load on any cable to B supplying the group of substations. The value of L may be readily used to compute the power losses which would be incurred in any cable which is then used to supply this group. Suppose R_{AB} denotes the yearly cost of losses which would be incurred in cable $A \rightarrow B$ at unit loading. Then in the determination of, for example, $f(A; BCD)$, the minimum of functions of the type

$$(A \rightarrow B) + f(B; CD L) + L^2 R_{AB}$$

is required.

As an illustration of this, the problem shown in Fig. A7 was re-solved taking these factors into account. Brief details of the costs used are as follows:-

- cable 0.3 sq. in. rated at 20 MVA, 33 kV

Cost £9 per yard total; equivalent annual cost
(20 year life) £0.784 per yard.

switchgear a cost of £5000 per circuit breaker, equivalent to £436 per annum.

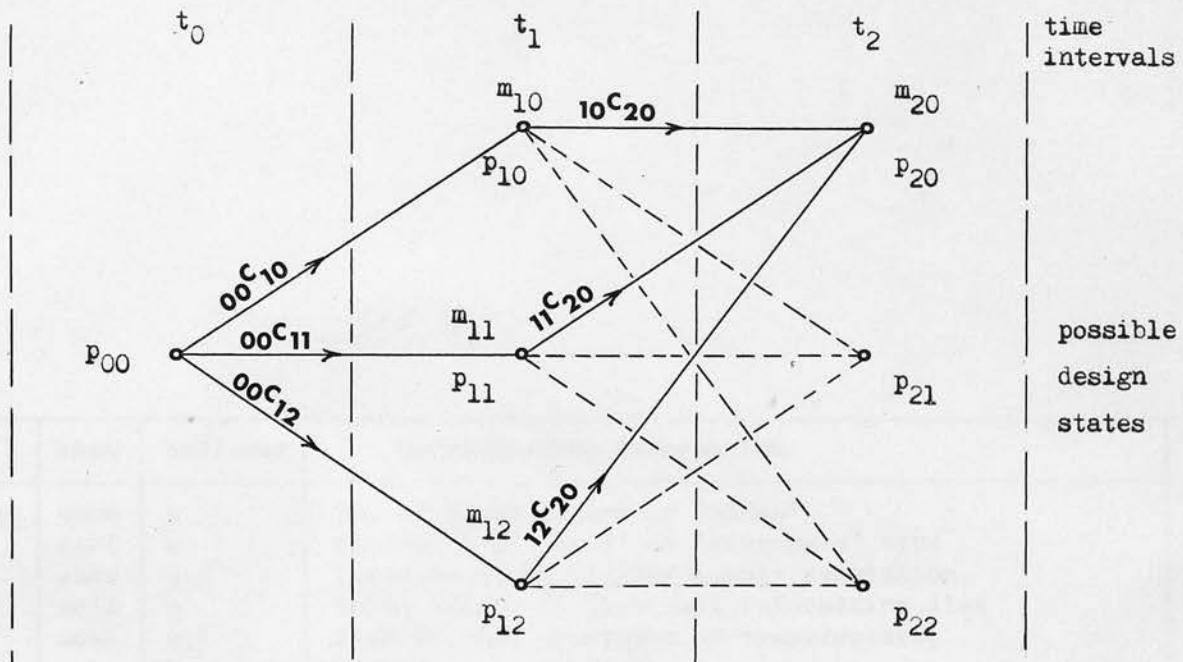
cable power losses 37.8 kW/1000 yards at 355 amps.

Assuming a cost of 1d. per kW hr., and average loads over a yearly interval of $\frac{1}{4}$ of the maximum figures specified for each substation (Fig. A9(a)),

this becomes £0.000838/yard/annum at 1 MVA maximum loading.

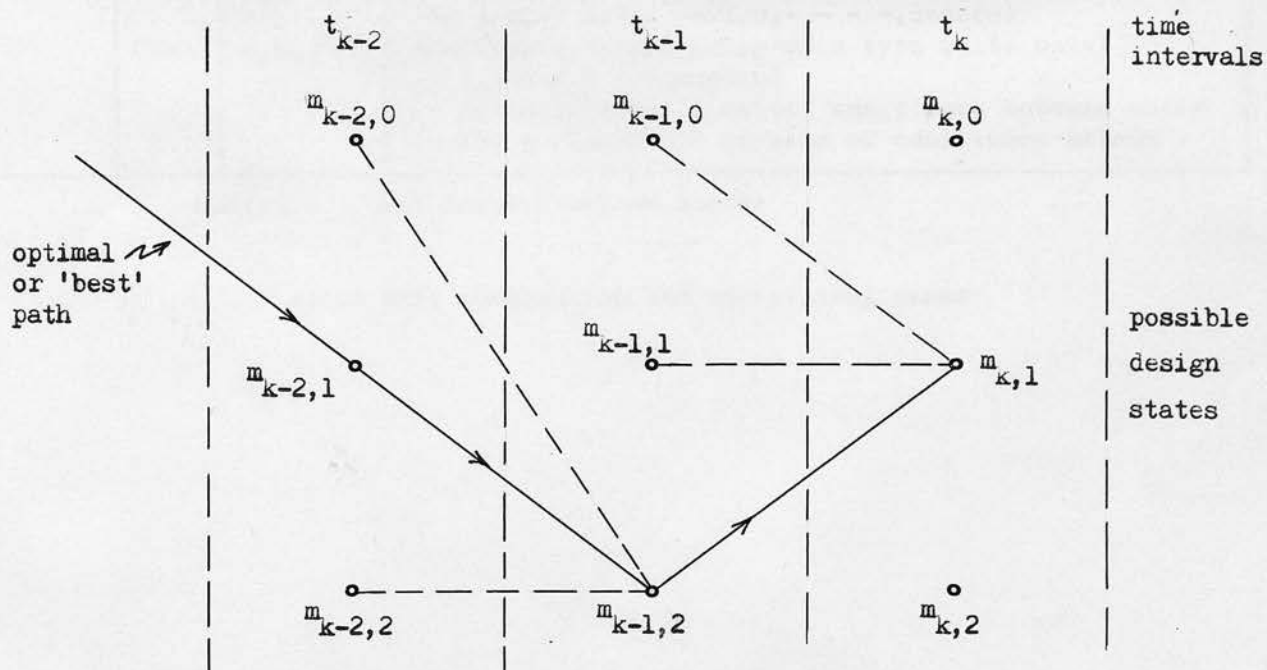
Fig. A9(a) gives the corresponding distance table, in units of 1000 yards, together with substation loadings (merely a re-scaled version of Fig. A7(a)). Fig. A9(b) gives the corresponding annual capital costs for cables and circuit-breakers, and Fig. A9(c) the annual costs of losses in each cable at 1 MVA loading. The stage by stage dynamic programming analysis is given in Fig. A9(d). (In this analysis the supply of system losses was ignored in the calculation of cable loadings.)

The solution turned out to be identical with that of the problem of Fig. A7(a), in spite of the inclusion of the costs of switchgear and system power losses. The cost of the power losses formed a significant portion of the total system costs, which might have been expected to favour a solution with more cable, but lower power losses. But this was balanced by the inclusion of switchgear costs, which would tend to demand the minimum number of separate spur units.



Basis of the optimal path algorithm

Figure A1



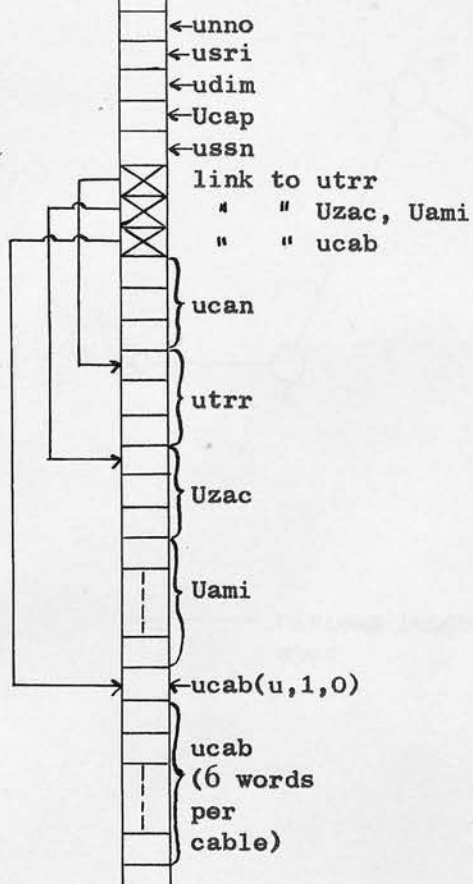
Nth best path algorithm, using best path as starting point

Figure A2

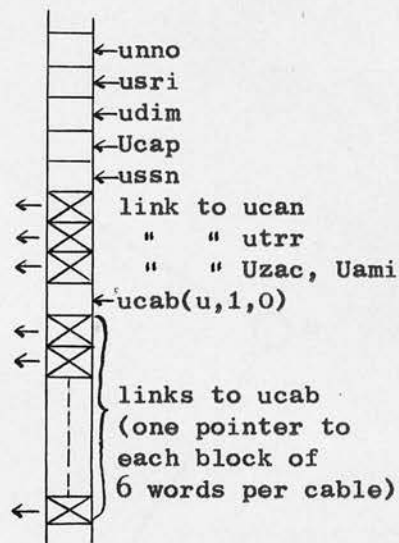
Type	Name	Suffices	Corresponding Information
<u>integer</u>	unno	u	No. of transformers or 'nodes'
	usri	u	Indicator of 'spur' or 'ring-main' type
	ussn	u	Location of 275(132)kV supply substation
	udim	u	Total length of this unit information list
	ucan	u,i	List of node locations of transformers, i=1,2,- - -,unno(u)
	utrr	u,i	List of transformer sizes
	ucab	u,i,j	List of details of unit cables ucab(u,1,0)=no. of cables; i=1,- - -,ucab(u,1,0) j=1,2 end connection nodes =3,4 switchgear details for cable ends =5,6 location and length of cable route details
<u>real</u>	Ucap	u	Yearly capital charges on unit
	Uzac	u,i	Admittances of all cables connecting load tr'f'rs to supply node. i=1,2,- - -,unno(u)
	Uami	u,i,j	Admittance matrix (ring-main type units only) i,j=1,2,- - -,unno(u) Below diagonal - actual admittance between nodes Above diagonal - inverse of admittance matrix

Suffix u = unit identification number

A3(a) Unit information and identifying names



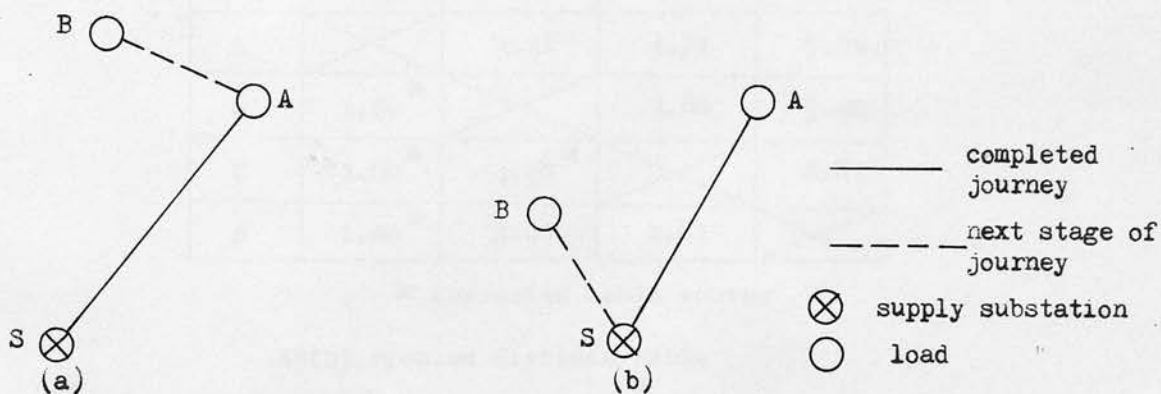
A3(b) Original unit store arrangement



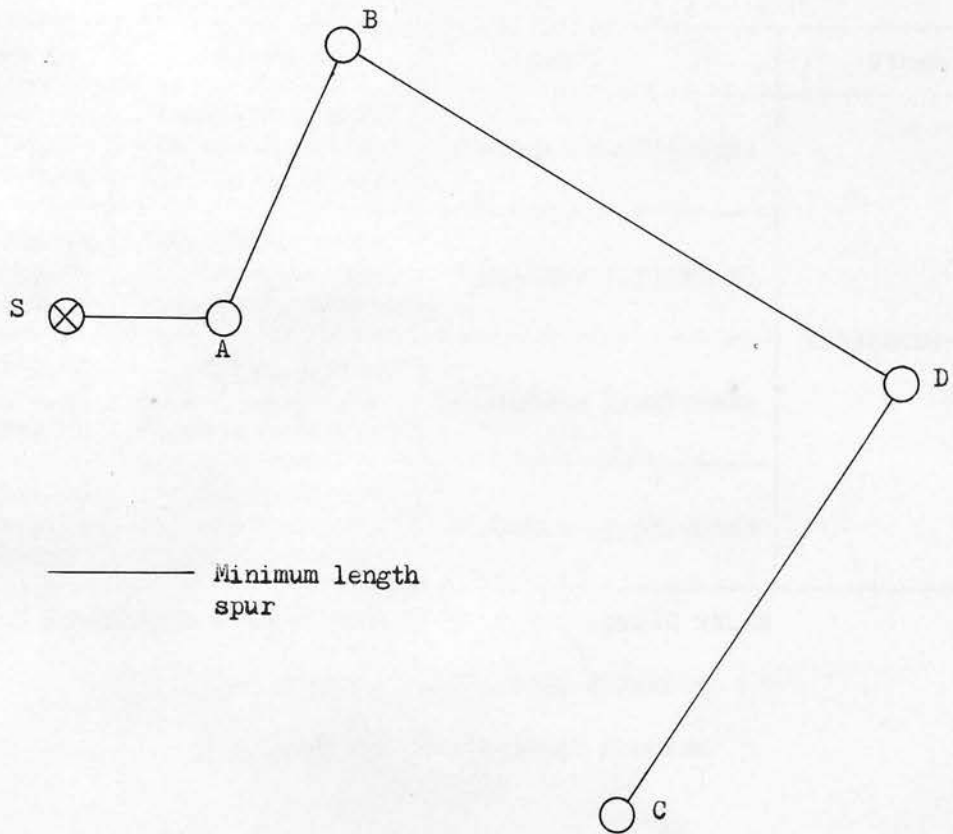
A3(c) Possible modified store arrangement

← ⊗ Indicates a word containing the address of further information

Storage of unit information
Figure A3



Introduction of corrected cable routes in the travelling salesman problem
Figure A4



A5(a) A four substation supply problem

	A	B	C	D
S	1.00	4.48	5.59	7.03
A	1.00	3.41	4.71	6.00
B	1.00*	3.41	4.60	3.90
C	1.00*	4.48*	5.59	2.81
D	1.00*	3.90	2.81	6.00

* Corrected cable routes

A5(b) Problem distance table

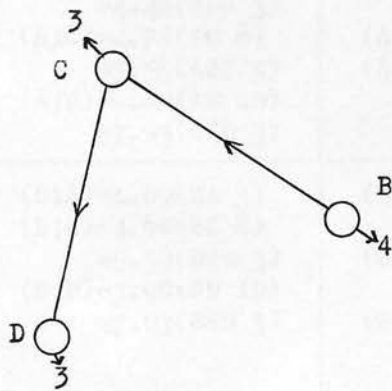
Stage 1	Stage 2	Stage 3	Stage 4
$f(A;B)=3.41$ $f(A;C)=4.71$ $f(A;D)=6.00$	$f(A;BC)=8.01(ABC)$ $f(A;BD)=7.31(ABD)$ $f(A;CD)=7.52(ACD)$	$f(A;BCD)=10.12(ABDC)$	$f(S;ABCD)=11.12$ $(SABDC)$
$f(B;A)=1.00$ $f(B;C)=4.60$ $f(B;D)=3.90$	$f(B;AC)=5.60(BCA)$ $f(B;AD)=4.90(BDA)$ $f(B;CD)=6.71(BCD)$	$f(B;ACD)=7.71(BDCA)$	
$f(C;A)=1.00$ $f(C;B)=4.48$ $f(C;D)=2.81$	$f(C;AB)=4.41(CAB)$ $f(C;AD)=3.81(CDA)$ $f(C;BD)=6.71(CDB)$	$f(C;ABD)=7.22(CDAB)$	
$f(D;A)=1.00$ $f(D;B)=3.90$ $f(D;C)=2.81$	$f(D;AB)=4.41(DAB)$ $f(D;AC)=3.81(DCA)$ $f(D;BC)=7.29(DBC)$	$f(D;ABC)=7.22(DCAB)$	

Solution = S → A → B → D → C ; length 11.12

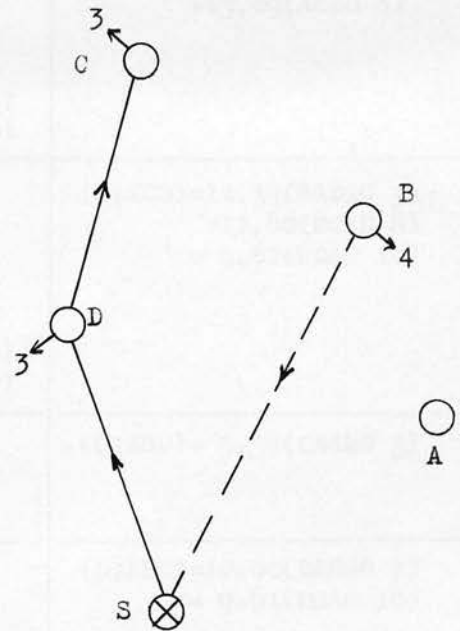
A5(c) Dynamic programming solution

A simplified 'laynewspur' problem

Figure A5



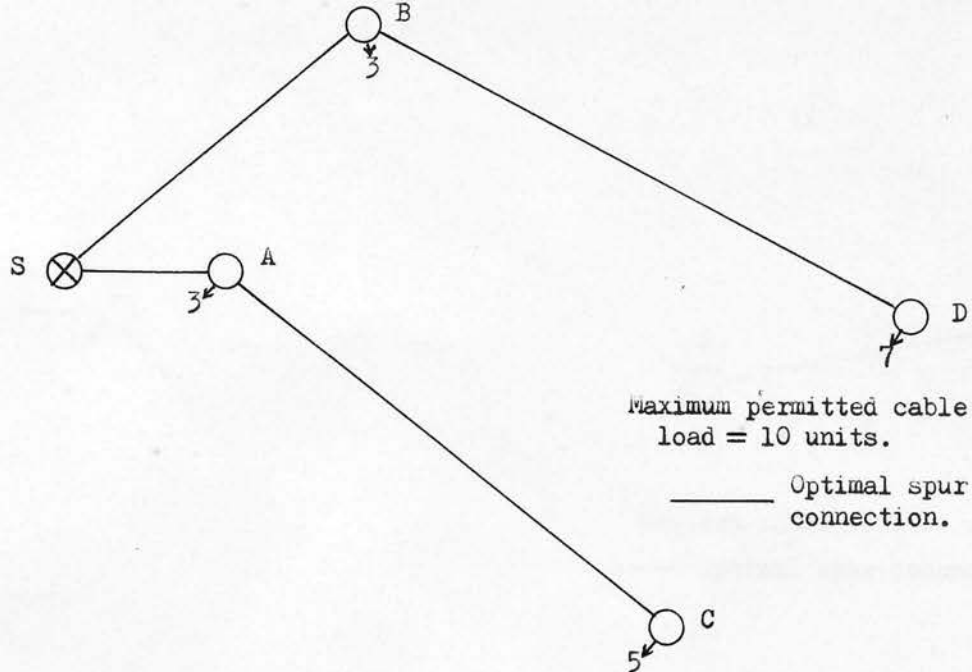
A6(a) Sub-system (B;CD 10)



A6(b) Sub-system (B;CD 4)

Substation sequences with differing initial cable loads

Figure A6



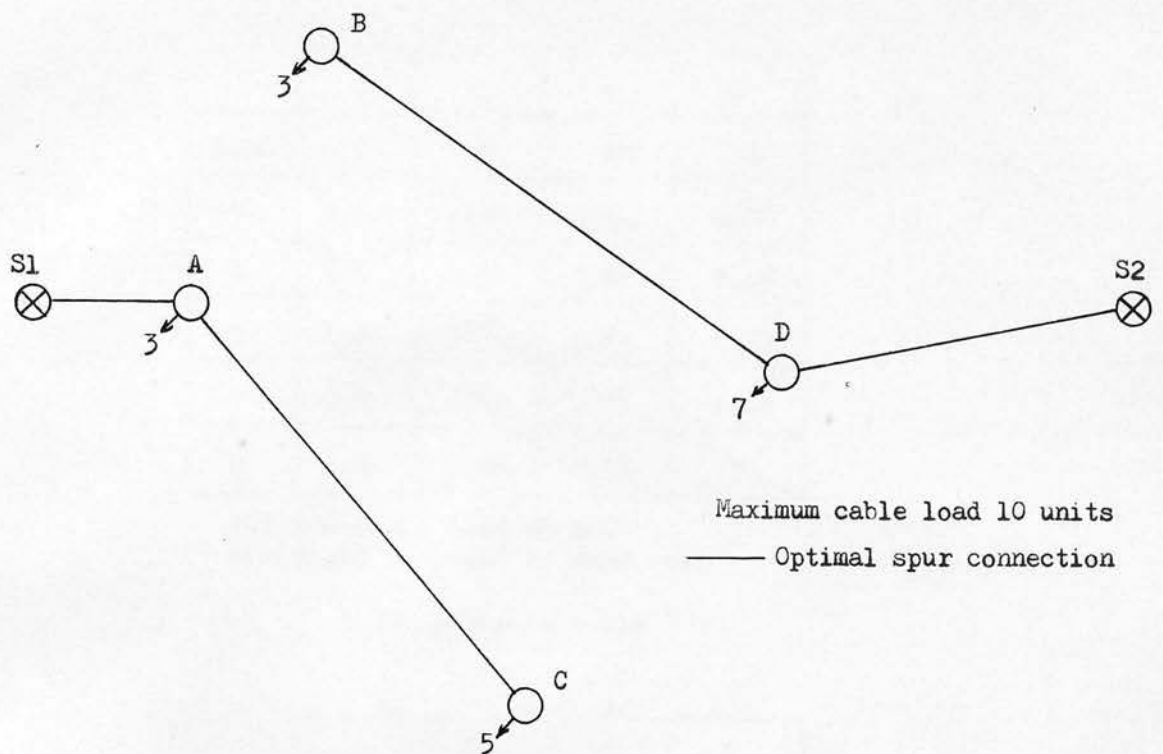
A7(a) Four substation problem with cable load restriction

Stage 1	Stage 2	Stage 3
$(A;B)=3.41(AB\ 6)$ $=4.48(ASB\ 3)$ $(A;C)=4.71(AC\ 8)$ $=5.59(ASC\ 3)$ $(A;D)=6.00(AD\ 10)$ $=7.03(ASD\ 3)$	$(A;BC)=9.00(ABSC\ 6)$ $=9.08(ASBC\ 3)$ $(A;BD)=8.38(ASBD\ 3)$ $(A;CD)=11.74(ACSD\ 8)$ $=12.62(ASCSD\ 3)$ $=11.59(ADSC\ 10)$	$(A;BCD)=13.97(ASBDSC\ 3)$ $=13.09(ACBD\ 8)$
$(B;A)=1.00(BA\ 3)$ $(B;C)=4.60(BC\ 8)$ $=5.59(BSC\ 3)$ $(B;D)=3.90(BD\ 10)$ $=7.03(BSD\ 3)$	$(B;AC)=5.71(BAC\ 3)$ $=5.60(BCA\ 8)$ $(B;AD)=7.00(BAD\ 3)$ $=4.90(BDA\ 10)$ $(B;CD)=11.63(BCSD\ 8)$ $=12.62(BSCSD\ 3)$ $=9.49(BDSC\ 10)$	$(B;ACD)=12.59(BADSC\ 3)$ $=11.60(BCAD\ 8)$ $=9.61(BDAC\ 10)$
$(C;A)=1.00(CA\ 5)$ $(C;B)=4.48(CB\ 5)$ $(C;D)=7.03(CSD\ 5)$	$(C;AB)=4.41(CAB\ 5)$ $(C;AD)=7.00(CAD\ 5)$ $(C;BD)=8.38(CBD\ 5)$	$(C;ABD)=9.38(CASBD\ 5)$
$(D;A)=1.00(DA\ 7)$ $(D;B)=3.90(DB\ 10)$ $=4.48(DSB\ 7)$ $(D;C)=5.59(DSC\ 7)$	$(D;AB)=4.41(DAB\ 7)$ $(D;AC)=5.71(DAC\ 7)$ $(D;BC)=9.08(DSBC\ 7)$	$(D;ABC)=10.00(DABSC\ 7)$ $=9.61(DBAC\ 10)$
Stage 4		
$(S;ABCD)=14.09(SACBD\ 8)$ Solution = $S \rightarrow A \rightarrow C$; $S \rightarrow B \rightarrow D$; length = 14.09		

A7(b) Extension of dynamic programming to two dimensions

A 'laynews spur' problem with cable load restriction

Figure A7



A8(a) Problem with two supply substations




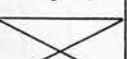
	A	B	C	D
Load	3	3	5	7
S	1.00(S1)	4.48(S1)	5.56(S2)	3.02(S2)
A	1.00	3.41	4.71	3.02*
B	1.00*	3.41	4.60	3.02*
C	1.00*	4.48*	5.56	2.81
D	1.00*	3.90	2.81	3.02

* Corrected cable routes

A8(b) Distance table - two supply substations

A 'laynewspur' problem with cable load restrictions and two supply substations





Figure A8

	A	B	C	D
Load	6	6	10	14
S	0.04	1.79	2.24	2.81
A		1.36	1.88	2.40
B	1.36		1.84	1.56
C	1.88	1.84		1.12
D	2.40	1.56	1.12	

Maximum cable load 20 MVA

Distances in units of 1000 yds.




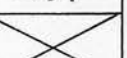
A9(a) Distance table

	A	B	C	D
S	0.750	1.840	2.193	2.640
A		1.068	1.475	1.883
B	*0.750		1.444	1.224
C	*0.750	1.444		0.878
D	*0.750	1.224	0.878	

* Corrected cable routes

Costs in units of £1000

A9(b) Equivalent annual costs of cables and switchgear

	A	B	C	D
S	0.34	1.50	1.88	2.36
A		1.14	1.58	2.01
B	*0.34		1.54	1.31
C	*0.34	1.54		0.94
D	*0.34	1.31	0.94	

* Corrected cable routes

Costs in units of £1

A9(c) Annual costs of system losses at 1 MVA cable loadings

Stage 1	Stage 2	Stage 3
$(A;B)=1.109(AB\ 12)$ $=1.894(ASB\ 6)$ $(A;C)=1.633(AC\ 16)$ $=2.381(ASC\ 6)$ $(A;D)=2.277(AD\ 20)$ $=3.103(ASD\ 6)$	$(A;BC)=3.822(ASBC\ 6)$ $=3.490(ABSC\ 12)$ $=3.482(ACSB\ 16)$ $(A;BD)=3.921(ASBD\ 6)$ $(A;CD)=4.736(ACSD\ 16)$ $=5.484(ASCSD\ 6)$ $=4.658(ADSC\ 20)$	$(A;BCD)=6.302(ASBDSC\ 6)$ $=6.099(ADSBC\ 20)$
$(B;A)=0.726(BA\ 6)$ $(B;C)=1.598(BC\ 16)$ $=2.381(BSC\ 6)$ $(B;D)=1.481(BD\ 20)$ $=3.103(BSD\ 6)$	$(B;AC)=2.469(BSAC\ 6)$ $=2.360(BCA\ 16)$ $(B;AD)=3.161(BSAD\ 6)$ $=2.243(BDA\ 20)$ $(B;CD)=4.701(BCSD\ 16)$ $=5.484(BSCSD\ 6)$ $=3.862(BDSC\ 20)$	$(B;ACD)=5.542(BSADSC\ 6)$ $=4.759(BCAD\ 16)$ $=3.950(BDAC\ 20)$
$(C;A)=0.762(CA\ 10)$ $(C;B)=1.499(CB\ 16)$ $=1.894(CSB\ 10)$ $(C;D)=3.103(CSD\ 10)$	$(C;AB)=1.907(CAB\ 10)$ $(C;AD)=3.161(CAD\ 10)$ $(C;BD)=3.921(CSBD\ 10)$	$(C;ABD)=4.683(CASBD\ 10)$ $=4.600(CBSAD\ 16)$
$(D;A)=0.762(DA\ 14)$ $(D;B)=1.271(DB\ 20)$ $=1.894(DSB\ 14)$ $(D;C)=2.381(DSC\ 14)$	$(D;AB)=1.907(DAB\ 14)$ $(D;AC)=2.469(DAC\ 14)$ $(D;BC)=3.822(DSBC\ 14)$ $=3.652(DBSC\ 20)$	$(D;ABC)=4.288(DABSC\ 14)$ $=3.740(DBSAC\ 20)$
Stage 4		
$(S;ABCD)=6.390(SBDAC)$ Solution = S \rightarrow A \rightarrow C ; S \rightarrow B \rightarrow D : annual cost £6,390		

Costs in units of £1000

A9(d) Dynamic programming solution

A 'laynewspur' problem with the inclusion of switchgear and system loss costs
Figure A9

Bibliography

- [1] Design of Housing-Estate Distribution Systems using a Digital Computer.
R. L. Grimsdale, P. H. Sinclare. Proc. IEE No. 33 PtA June, 1960 pp 295-305.
- [2] Economic Design and Operation of Power Systems.
U. G. Knight. Electrical Review, 26th July, 1963 pp 124-30.
- [3] Calculation of the Optimum Equivalent Parameters in a Distribution Network.
J. V. Cox, U. G. Knight, A. K. Rhodes. Paper offered at Conference on Digital Computation for Electrical Power Systems. Queen Mary College. September, 1963.
- [4] The Logical Design of Electrical Networks using Linear Programming Methods.
U.G. Knight. Proc IEE No. 33 PtA. June, 1960 pp 306-14.
- [5] Outline of an Algorithm for Integer Solutions to Linear Programs.
R. E. Gomory. Bull. Am. Math. Soc., Vol. 64, 1958, pp 275-8
- [6] Design of Electricity Distribution Networks - A Heuristic Approach.
R. M. Burstal. University of Birmingham (Department Engineering Production) - Interim report 3rd March, 1964.
- [7] Methods for Integer Linear Programming of Certain Nonlinear Problems in Optimal Design.

- I. B. Motskus. Izvestia Akd. Nauk. U.S.S.R. Energetica i Transport No. 4 July-August, 1963.
- [8] Electric Power Systems Planning Using Linear Programming.
J. Bergsman. IEEE Trans (Military Electronics) VMIL-8
No. 2 April, 1964 pp 59-62.
- [9] Economic Problems in Electric Power System Planning.
J. Bergsman. Publication EEP-6, Stanford University,
July, 1963.
- [10] Digital Computer Applied to Electrical Distribution
System Planning.
E. A. Strausser AIEE Trans V77 pt 3 February, 1959 pp
1244-50.
- [11] An Introduction to the Study of System Planning by
Operational Gaming Models.
J. D. Dillard, H. K. Sels. AIEE Trans pt 3 December,
1959 pp 1284-90.
- [12] Model for Transmission Planning by Logic.
C. J. Baldwin, C. A. De Salvo, C. H. Hoffman, W. S. Ku.
AIEE Trans V78 Pt 3 February, 1960 pp 1638-43.
- [13] Distribution System Planning Through Optimised Design.
R. F. Lawrence, D. N. Reys, A. D. Patton. AIEE Trans
V79 Pt 3. June, 1960 pp 199-213.
- [14] Application of Computer Techniques in Power System
Operation and Design .
Markovich, Taft, Savalov, Benikov, Tsukernik. Elec-
trichestvo No. 11. 1960 pp 1-8.
- [15] Results of Two Year Study of Long-Range System Planning
by Simulation.

- C. H. Hoffman. 19th CIGRE C.Rn 306 May, 1962.
- [16] System Optimisation Techniques.
L. K. Kirchmayer. Electrical Engineering Vol. 81 No. 8
August, 1962 pp 618-22.
- [17] Computer Program for Automatic Transmission Planning.
J. L. Whyson, R. Vram, H. E. Brown C. W. King, C. A. De
Salvo. AIEE Trans Vol. 81 pt 3 February, 1963 pp 774-81.
- [18] Automated Distribution System Planning.
R. F. Lawrence, R. E. Montmeat, A. D. Patton, D. Wappler.
IEEE Trans (Power App. and Systems) V83 No. 4 April, 1964
pp 311-16.
- [19] Secondary System Distribution Planning for Load Growth.
R. M. Weblor, M. W. Gangel, G. K. Carter, A. L. Zeman,
R. C. Ender. IEEE Trans (Power App. and Systems) No. 69
December, 1963 pp 908-27.
- [20] The Applications of Methods of Mathematical Programming
to the Solution of the Problem of Perspective
Development of Electrical Power Systems.
P. M. Markovich, V. P. Brailov, B. I. Denicov. Izvestia
Akd. Nauk U.S.S.R. Energetika i Avtomatika No. 6 November-
December, 1962 pp 11-16.
- [21] Digital Computers in Distribution Network Studies.
P. Gaussens - Paper at Conference on Digital Computation
for Electrical Power Systems Queen Mary College.
September, 1962.
- [22] An Algorithm for the Solution by Digital Computer of
Three Power Problems of Optimisation within Time.
E. B. Tsetkov. Izvestia Akd. Nauk. U.S.S.R.

Energetika i Avtomatika No. 1 January-February, 1962
pp 54-62.

- [23] Dynamic Programming (book).

R. Bellman. Princeton N.J. 1957.

- [24] Algorithm for Finding the Shortest Paths Between Pairs
of Points in a Network.

F. C. Bock. Phase report No. 1 Armour Research Foundation
Project EO84 June, 1957.

- [25] Mathematical Programming (book) (Shortest Path pp47-48).

S. Vajda. Addison-Wesley Pub. Co. Inc. 1961.

- [26] A Method for the Solution of the Nth Best Path Problem.

F. C. Bock. J. Assoc. Com. Mach. 6 1959 pp 506-14.

- [27] On kth Best Policies.

R. Bellman, R. Kalaba. J. Soc. Ind. and App. Maths.
V8 December, 1960 pp 582-8.

- [28] Computing the N Best Loopless Paths in a Network.

S. Clarke, A. Krikorian, J. Rausen. J. Soc. Ind. and
App. Maths V11 December, 1963 pp 1096-1102.

- [29] The Addition of Vectors.

J. L. Egginton. Electrical Times Vol. 119, No. 3097,
15th March, 1951, pp 439-42.

- [30] The Application of Linear Algebra to Network Systems.

J. Hewson, A. K. Rhodes. C.E.G.B. paper CD/DEV/1
July, 1963.

- [31] Atoms and Lists.

P. M. Woodward, D. P. Jenkins. Computer Journal, Vol.
4, 1961-2 pp 47-53.

- [32] The MAC System: The Computer Utility Approach.
R. M. Fano. IEEE Spectrum, Vol. 2, No. 1, January,
1965 pp 56-64.
- [33] Travelling Salesman Problem by Dynamic Programming.
R. Bellman, J. Assoc. Com. Mach. January, 1962.
- [34] The Long-Term Design of Electrical Power Distribution
Systems.
J. V. Oldfield, T. Lang. Proc. 1965 P.I.C.A. Conference
of IEEE pp 671-82.